Lower bounds for Quantum Computation and Communication

by

Ashwin V. Nayak

B.Tech. (Indian Institute of Technology, Kanpur) 1995

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Umesh Vazirani, Chair
Professor Alistair Sinclair
Professor Raymond Chiao

Fall 1999

The dissertation of Ashwin V. Nayak is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 1999

Lower bounds for Quantum Computation and Communication

by

Ashwin V. Nayak

Abstract

Lower bounds for Quantum Computation and Communication

by

Ashwin V. Nayak

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Umesh Vazirani, Chair

The description of the state of an n-bit quantum system requires $2^n - 1$ complex numbers. This exponentially large information capacity of quantum states has been exploited in recent results showing both exponential speed-up, and exponential savings in communication costs in solving certain problems using quantum computers. In this dissertation, we establish limitations on the ways in which the exponentially many degrees of freedom hidden in quantum states may be accessed. More specifically, we give tight bounds for random access codes, which allow us to encode classical information using quantum bits such that only a small portion of the encoded information may be recovered via a measurement. This also sheds light on the power of computing with a finite number of quantum bits—using these techniques, we show an exponential size lower bound for quantum finite automata for a problem which can be solved on a linear size classical automaton. We then consider the complexity of solving certain problems in the quantum black-box model, an information theoretic model that has been a rich source of insights into the nature of quantum computation. We derive nearly optimal lower bounds for several problems in this model, including that of approximating the median. We also give new, optimal algorithms for approximate medians and other order statistics.

Professor Umesh Vazirani
Dissertation Committee Chair

To my parents.

Contents

List of Figures

Acknowledgements

I would like to thank Umesh Vazirani for introducing me to the fascinating world of quantum computing. He has been the quintessential advisor in all aspects of my research. I have been very fortunate to have been guided also by Alistair Sinclair, especially during my early forays into research at Berkeley. I am also thankful to Professor Raymond Chiao for advising me on the physical aspects of quantum computation.

Parts of Chapter 3 of this thesis represent work done in collaboration with Andris Ambainis, Amnon Ta-Shma and Umesh Vazirani, and much of the research presented in Chapter 4 is joint work with Felix Wu. I am thankful to my collaborators for stimulating my interest in the problems studied there.

Finally, I would like to thank my dissertation committee for their detailed comments on earlier drafts. I am also grateful to Andris Ambainis and Anupam Gupta for reading of parts of the draft and giving me valuable feedback.

Chapter 1

Introduction

The study of the intrinsic complexity of computational problems is based on the modern version of the Church-Turing Thesis, which says that all reasonable models of computation can be simulated efficiently by a probabilistic Turing machine. In 1982, Feynman raised the question as to whether quantum physics could be simulated efficiently on conventional computers [43]. The issue here is that an n-component quantum system (such as a collection of n nuclear spins) in general exists in a *superposition* of all its observable configurations (all the 2^n possible combinations of "up" and "down" in the case of spins), and so any straightforward method of simulating its behaviour suffers an exponential slowdown. Feynman suggested computers with quantum mechanical parts as a way of overcoming this limitation, thus implicitly challenging the very foundations of computer science. The task of formalising these ideas was taken up only a little later [36, 37], and a sequence of results earlier this decade due to Bernstein and Vazirani [16], and Simon [72], culminating in the polynomial time algorithms for Discrete Logarithms and Factoring due to Shor [71] convincingly demonstrated the potential of quantum mechanical computers to provide exponential speed-up over classical ones. Since then, the power of quantum physical primitives in computing has been established in many different contexts. In 1995, Grover discovered an algorithm for searching an unordered database that is quadratically faster than possible classically [46]. In 1997–98, it was shown that certain communication tasks can be performed by exchanging significantly (even exponentially) fewer quantum bits as compared to classical bits [24, 9, 68]. More recently, Watrous proved that PSPACE has constant-round quantum interactive proof systems [75]. Tremendous effort has also been invested in the experimental realisation of quantum computation and communication. There are numerous

proposals for experimental systems such as those based on ion traps [30] and nuclear magnetic resonance (NMR) [45, 33]. Simple quantum algorithms have also been implemented using NMR (see, for example, [29, 53]) and quantum key distribution protocols have been tested over increasingly larger distances [51].

The computational and information-theoretic advantages of using quantum mechanical primitives can perhaps be traced back to the phenomena of entanglement and interference in the exponential size Hilbert space in which quantum states reside. Consider, for example, the following situation. Alice has an n-bit secret x that Bob wishes to know. However, she only agrees to give cryptic answers to questions asked by Bob: on a query y, she replies with $x \cdot y = \sum_i x_i \, y_i \pmod 2$. Classically, every reply to a question by Bob reveals only one bit of information about x, and so Bob has to query Alice n times to learn x. On the other hand, by querying Alice with a superposition of points, Bob can determine x by asking a *single* question! He queries Alice with a uniform superposition of points $2^{-n/2} \sum_y |y\rangle$ and inverts the phase of points where $x \cdot y = 1$ to get the state $|\phi_x\rangle = 2^{-n/2} \sum_y (-1)^{x \cdot y} |y\rangle$. The states $\{|\phi_x\rangle\}$ are all mutually orthogonal for different x, so Bob can identify x by making a suitable measurement of his state. This was the basis of a result of Bernstein and Vazirani [16] which gave the first evidence of the super-polynomial speed-up possible with quantum computers.

As seen in the example above, results showing the superiority of quantum computing seem to defy all conventional intuition about computation. It is therefore of crucial importance to understand the limits of the power of quantum computation. One obstacle to proving strong results in this direction is that P $\subseteq$ BQP $\subseteq$ PSPACE [16, 1][1]. Thus, showing that any non-trivial problem is not in BQP would imply a separation of P from PSPACE, a long standing open problem in complexity theory. However, we can still get insights into the limitations of quantum computing by looking at restricted models such as the *black-box model*. In this model, information about the input is provided to an algorithm by an *oracle*. An oracle may be seen as a subroutine whose code is inaccessible and expensive to run, and we would like to solve problems by making the minimum number of invocations of the subroutine.

The black-box model allows us to formalise and evaluate strategies for solving

[1] P is the class of problems decidable in polynomial time on a deterministic Turing machine and PSPACE is the class decidable with polynomial amount of space. BQP is the class of problems that can be solved on a quantum Turing machine in polynomial time and with bounded probability of error.

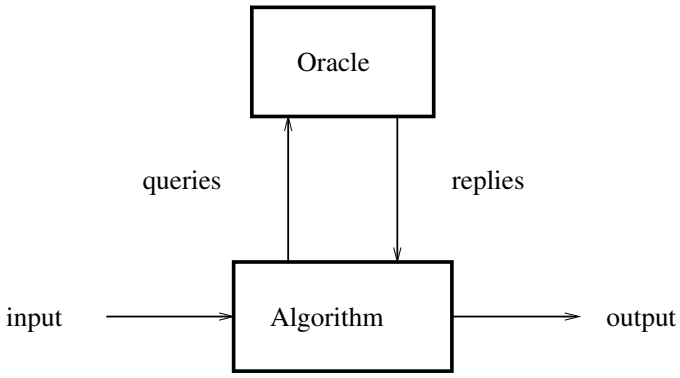

Figure 1.1: *In* black-box *computation, an algorithm communicates with an* oracle *to solve problems. The input to the oracle subroutine is called the* query *and its output, the* reply*.*

problems. For example, we could model "brute-force search" in the case of the satisfiablility problem by assuming that we are given an oracle that tells us whether a given assignment of truth values to the variables in a boolean formula (hidden from the algorithm) satisfies it or not. The aim would be to determine satisfiability of the hidden formula by querying the oracle as few times as possible. The black-box model is closely related to the two-party communication model, where two "players" wish to compute a joint function of their private inputs with as little communication as possible. Computing in this model is, in effect, a communication game between the algorithm and the oracle, where the communication follows a specific query-reply format (see Figure 1.1). The model thus lies at the intersection of quantum computing and quantum communication and reinforces the need to investigate the properties of quantum physical primitives from both perspectives.

In this thesis, we first study the information theoretic properties of quantum states in the context of the general quantum communication model. We consider different ways of encoding classical information in low dimensional quantum states and and prove tight limits on how much compression can be achieved via such encoding. Our study finds an unexpected application in the realm of computing—that of showing a size lower bound for quantum finite automata. Next, we concentrate on the black-box model itself, and consider the complexity of solving certain problems for which quantum algorithms provably more efficient than classical ones were discovered recently. We show that the algorithms are nearly optimal in some cases and present new, optimal algorithms in others. We elaborate on these results below.

1.1 Results on quantum encoding and communication

Quantum physics provides us with a new means of transmitting information—via the exchange of *quantum bits* (or simply *qubits*, two-state quantum systems such as polarised photons). The state of a collection of n qubits is given by a unit vector (a *superposition*) in the n-fold tensor product of the two-dimensional complex Hilbert space generated by the two (basis) states. Thus, the state is a unit vector in a 2^n-dimensional vector space. Now consider a situation where one party, Alice, wishes to send some message (a bit string) to another party, Bob. Could she exploit this fact to encode her message into much fewer quantum bits such that Bob could recover the message without incurring much error? While the exponential description of quantum states offers an attractive possibility for encoding information, the amount of information that can be extracted from them is limited by the nature of the measurement process. A fundamental result from quantum information theory, Holevo's theorem [50], says that n qubits cannot be used to transmit more than n bits of classical information. Given this result, it is tempting to conclude that the exponentially many degrees of freedom latent in the description of a quantum system must necessarily stay hidden or inaccessible. However, one can convey information in highly non-obvious ways via the exchange of quantum states. For example, Ambainis, Schulman, Ta-Shma, Vazirani and Wigderson [9] show how to deal cards over a quantum phone exponentially more efficiently than is possible classically. They show that two communicating parties may pick disjoint subsets of $\{0, \ldots, n-1\}$ of size $\sqrt{n}$ almost uniformly at random by transmitting only $O(\log n)$ quantum bits, whereas $\Omega(\sqrt{n})$ bits are required classically. The properties of quantum states are thus very subtle, and a deeper study of quantum information is called for.

Holevo's theorem bounds the mutual information between two classical random variables X and Y, where Y is obtained by making some measurement on a quantum encoding of X. It thus forms a very basic tool for proving lower bounds in communication scenarios. Indeed, it has been used to prove lower bounds for the communication complexity of problems such as computing the inner product [55, 31]. We re-examine the problem of encoding classical information into states of as few quantum bits as possible, and give a tight analysis of the probability of decoding it correctly [61]. This gives us an alternative proof of the fact that we cannot transmit classical data using much fewer quantum resources.

Theorem 1.1.1 *Let X be a random variable over bit strings which are encoded into states*

over m qubits and let $P(X, d)$ denote the net probability of the d most likely strings in the sample space of X. If Y is any random variable obtained by making some measurement of the encoding of X, then for any decoding function $\mathcal{D}$, $\Pr[\mathcal{D}(Y) = X] \leq P(X, 2^m)$.

In typical situations, Holevo's bound is applied by converting a statement about the probability of correct decoding into a statement about mutual information. Our bound thus obviates the need for a translation of in-probability statements into statements in terms of entropy in these cases, also giving better bounds than Holevo's theorem in the process. It is perhaps worth mentioning that the proof of Holevo's theorem (which is closely related to the strong subadditivity property of von Neumann entropy) is quite involved while our result is fairly transparent.

Next, we consider a situation slightly different from that in Holevo's theorem, where the recipient of the encoding of a certain number of classical bits is interested in only *one*, *a priori* unknown bit of the original set. In this scenario, Holevo's theorem doesn't apply, since the measurement the recipient makes to extract one bit could potentially destroy some or all of the remaining encoded information. This allows us to encode, for example, classical two bits into *one* qubit such that any one of the bits can be retrieved with probability ≈ 0.85 by choosing one of two measurements appropriately. Such compression is not possible classically [8]. More generally, we can define an (n, m, p)-*random access encoding* as a function f that maps n-bit strings into states over m qubits such that, for every $i \in \{1, \ldots, n\}$, there is a measurement $\mathcal{O}_i$ with outcome 0 or 1 that has the property that for all $x \in \{0, 1\}^n$,

$$\Pr\left[\mathcal{O}_i(f(x)) = x_i\right] \geq p.$$

There is no *a priori* reason to rule out the existence of random access encoding of c^n bits into n quantum bits for a constant $c > 1$, although this is not possible classically. In fact, even though $\mathbb{C}^k$ can accommodate only k mutually orthogonal unit vectors, it can accommodate c^k almost mutually orthogonal unit vectors (i.e., vectors such that the inner product of any two has absolute value less than, say, $\frac{1}{10}$) for some $c > 1$. Could this be converted into a random access encoding? Such quantum encoding, if possible, would serve as a powerful primitive in quantum communication. For instance, it would be possible to compress the contents of an entire telephone directory into a few quantum bits such that the recipient of these qubits could, via a suitably chosen measurement, look up any *single* telephone number of his choice. However, we show that the above intuition is ill-

founded [8, 61]. Despite the promise shown by quantum encodings, random access codes require a linear number of quantum bits. In fact, our lower bound asymptotically matches the *classical* upper bound we give for the problem.

Theorem 1.1.2 *For any $p > \frac{1}{2}$, there is a classical (n, m, p)-random access encoding with $m = (1 - H(p))n + O(\log n)$. Moreover, any quantum (n, m, p)-random access encoding has $m \geq (1 - H(p))n$.*

Interestingly, our lower bound uses a general principle based on Holevo's theorem, although the theorem itself does not apply to this situation. We give a novel application of this principle to showing a lower bound on the size of one-way quantum finite automata (QFAs) [61].

Theorem 1.1.3 *Let L_n be the language $\{w0 \mid w \in \{0,1\}^*,\ |w| \leq n\}$. Then,*

1. *L_n is recognised by a deterministic finite automaton of size $O(n)$,*
2. *L_n is recognised by some QFA, and*
3. *Any QFA recognising L_n with some constant probability greater than $\frac{1}{2}$ has $2^{\Omega(n)}$ states.*

QFAs model quantum computers that work with a finite set of quantum bits. They have drawn much interest not only because their study provides insight into the nature of quantum computation, but also because they closely model the capability of currently feasible experimental quantum computers.

Kondacs and Watrous [54] introduced a model of QFA that allowed limited observations during the computation process, and showed that not every language recognised by a (classical) deterministic finite automaton (DFA) is recognised by such QFAs. On the other hand, Ambainis and Freivalds [7] showed how we may exploit the exponential resources afforded by quantum states to design QFAs for certain problems that are exponentially smaller than the corresponding classical automata. It remained open whether, for any language that can be recognised by a one-way finite automaton both classically and quantum-mechanically, a classical automaton can be simulated efficiently by a QFA. In [8] we gave a partial answer to this question by demonstrating that the requirement of reversible evolution seriously limits the efficiency of the QFAs of [54]. The arguments of [54, 8] break down when arbitrary quantum operations (in particular, measurements)

on a fixed set of quantum bits are allowed, since the evolution of their state is no longer reversible. Our result, Theorem 1.1.3, settles this issue by extending (and strengthening) the result of [8] for general QFAs as well by using completely different information theoretic techniques.

1.2 Results on quantum black-box complexity

Bennett, Bernstein, Brassard and Vazirani [13] first studied the problem of whether NP-complete problems can be solved efficiently on a quantum computer. Despite parallelism and the potential for interference inherent in quantum computation, it was suspected that BQP does not contain NP. Bennett *et al.* gave an exponential lower bound for an NP problem in the quantum black-box model, thus giving formal evidence in support of this belief. Underlying the problem considered by them is the fundamental search problem. Their problem may be viewed as searching for a "one" (e.g., a satisfying assignment for a boolean formula in n variables) in a database of 2^n bits (the truth table for the formula), or as distinguishing the case when there are no 1s from the case when there is at least one. This problem was later studied by Grover [46], who gave an $O(2^{n/2})$ query algorithm for it, thus showing that the lower bound of [13] is optimal. More generally, there is an $O(\frac{1}{\sqrt{\epsilon}})$ query algorithm that distinguishes the all zeros case from the case when there are an ϵ fraction of ones [20].

A related problem is that of distinguishing a fraction half of ones from a fraction $\frac{1}{2}(1+\epsilon)$ of ones. In 1996, Grover gave an $O(\frac{1}{\epsilon})$ quantum algorithm for this problem [47, 48] in the context of approximating the median of a sequence of numbers. This is quadratically better than possible on a classical computer. (The problem is closely related to that of telling a random coin with bias δ from an unbiased coin. Classically, it takes $\Theta(\frac{1}{\delta^2})$ coin tosses to identify which of the two coins one has.) However, it was open whether this algorithm could be improved upon. In particular, the hybrid argument of [13] (see also [73]) yields a lower bound of $\Omega(\frac{1}{\sqrt{\epsilon}})$, whereas Grover's algorithm was suspected to be optimal. We prove that this is indeed the case by showing a lower bound of $\Omega(\frac{1}{\epsilon})$ queries for the problem.

Theorem 1.2.1 *Let $\epsilon \geq \frac{2}{n}$. Let $X = (x_0, \ldots, x_{n-1}) \in \{0,1\}^n$ such that either $|X| = \sum_i x_i = \frac{n}{2}$ or $|X| = (1+\epsilon)\frac{n}{2}$. Given X as an oracle, any quantum algorithm that decides*

correctly whether $|X| = \frac{n}{2}$ *or* $|X| = (1+\epsilon)\frac{n}{2}$ *makes at least* $\Omega(\frac{1}{\epsilon})$ *queries to* X*.*

Our result is based on the *polynomial method*, which was recently introduced to quantum complexity theory by Beals, Buhrman, Cleve, Mosca and de Wolf [11]. They show that the acceptance probability of a quantum algorithm making T queries to a boolean oracle can be expressed as a real multilinear polynomial of degree at most $2T$ in the oracle input. Thus, if an algorithm computes a boolean function of the oracle input with probability at least $\frac{2}{3}$, the corresponding polynomial *approximates* the function to within $\frac{1}{3}$ at all points in the boolean hypercube. So, by proving a lower bound on the degree of polynomials approximating the boolean function, we can derive a lower bound on the number of queries T the quantum algorithm makes. We cannot, however, follow this route directly for the problem of approximating the median, since the restriction of the problem to boolean inputs does not yield a well-defined function. Nonetheless, the restriction *does* yield a boolean relation. The main component of our result is thus a degree lower bound for polynomials that "approximate" symmetric boolean relations. (Section 4.2 contains a precise statement of the lower bound.) This degree bound generalises a result due to Paturi [63] and also gives lower bounds for the problems of approximating the kth smallest element or the mean of a sequence of numbers, and approximately counting the number of ones of a boolean function.

We then present a new, optimal algorithm for approximating the kth-smallest element. This yields an $O(\frac{1}{\epsilon})$ query algorithm for approximate medians, thus improving over Grover's algorithm [47, 48]. (The complexity of Grover's algorithm is $\tilde{O}(\frac{1}{\epsilon}\log M)$, when the numbers are drawn from a domain of size M.) The algorithm is a natural generalization of the minimum finding algorithm discovered by Dürr and Høyer [39]. The basic technique is that of randomised divide and conquer using which we reduce the problem to that of uniform sampling and approximate counting.

Both the lower and the upper bounds we obtain for computing order statistics such as the median hold also in the *comparison tree model*, which focuses on the number of comparisons between the input elements required to solve a problem. As a corollary, we obtain optimal comparison algorithms for selecting the kth-smallest element on a quantum computer.

Theorem 1.2.2 *Any comparison tree quantum algorithm that computes the* k*th-smallest element of a list of* n *numbers makes* $\Omega(\sqrt{k(n-k+1)}\,)$ *comparisons. Moreover, there is a*

quantum algorithm that solves this problem with $O(\sqrt{k(n-k+1)}\,)$ *comparisons.*

An $O(\sqrt{n}\,)$ comparison algorithm for the minimum was already known [39]. Our algorithm stands in interesting contrast to the classical case, where $\Theta(n)$ comparisons are required for any k [19].

1.3 Organisation of the text

The rest of the dissertation is arranged as follows. Chapter 2 describes the formalism of quantum information processing and the models we will be working with. Chapter 3 deals with the problems in quantum encoding and communication mentioned above and some of their ramifications. Chapter 4 focuses on the quantum black-box complexity of several problems of a statistical nature alluded to above. Some backgroud material from information theory and the theory of approximations necessary for our results is summarised in Appendix A. The appendix also contains the proofs of some claims made in Chapter 4.

Chapter 2

Models for quantum information processing

In this chapter, we briefly describe the formalism of quantum computation and communication. We first define quantum circuits and then move to the two-party quantum communication model. In the end, we focus on quantum computers that work with finite space, namely quantum finite automata. This involves the notion of a *density matrix* and some of its properties; these are given in detail in Section A.1 of the appendix.

2.1 Quantum oracle circuits

We now make precise the model of quantum computation and what it means to compute with access to an oracle in this context. We will discuss *quantum circuits*, since these are particularly convenient to work with. Quantum networks, the precursors to circuits, were introduced by Deutsch [37]. Yao [78] singled out quantum circuits as a special subclass of the network model and showed their equivalence in computational power to the quantum Turing machine model of Deutsch [36] and Bernstein and Vazirani [16]. Aharonov, Kitaev and Nisan [2] introduced a more general model of quantum circuits that is more appropriate in contexts such as computing in the presence of noise [70, 3] or with limited space [76].

Quantum circuits may be defined in a fashion analogous to classical boolean circuits. They consist of a sequence of *wires* and *quantum gates*. Each wire carries a *quantum bit* (or *qubit*), which is physically a two-state quantum system, such as the spin of an atomic

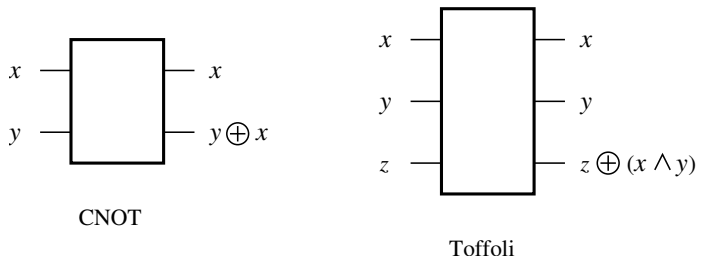

Figure 2.1: *Some classical reversible gates useful in quantum computation.*

nucleus. The state of a qubit is in general a *superposition* of its basis states, which are labelled as 0 and 1. Formally, this is a unit vector in a two-dimensional complex Hilbert space. Similarly, the joint state of the different wires is a unit vector in the Hilbert space corresponding to the tensor product of the complex spaces of the sequence of wires. The state of a quantum system is usually expressed in the Dirac "bra-ket" notation. For example, a superposition of n quantum bits is written as $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$. Here, the vectors $\{|x\rangle\}$ are the basis vectors of the 2^n-dimensional tensor product of n two-dimensional complex Hilbert spaces, and α_x, also called the *amplitude* of $|x\rangle$, is the component of the state vector along the basis element $|x\rangle$. The amplitudes satisfy the property that $\sum_x |\alpha_x|^2 = 1$. We also use the notation $|\phi\rangle$ to denote superpositions that are not necessarily basis states. It may also be understood to mean that the superposition is being represented in column vector form. The notation $\langle\phi|$ is used for the linear functional that maps a vector to its inner product with the superposition $|\phi\rangle$. It may also be understood as the conjugate transpose of the column vector $|\phi\rangle$.

The quantum gates are drawn from a small set of gates *universal* for quantum computation (see, e.g., [10]). The gates are described by local unitary operations, i.e., by linear transformations U that act as identity on all but a small (constant) number of the quantum bits (or wires), and satisfy the property that $UU^\dagger = I$, where $U^\dagger$ is the adjoint of U. In matrix representation, $U^\dagger$ is the conjugate transpose of the matrix U. It is convenient to describe the gates by specifying the list of wires on which the gate acts along with a unitary operator on those wires.

Examples of quantum gates are single qubit rotation, Hadamard gate, controlled not and controlled-controlled not (or Toffoli gate). The latter two are *reversible* versions of classical boolean gates and are depicted in Figures 2.1. The rotation gate rotates the basis

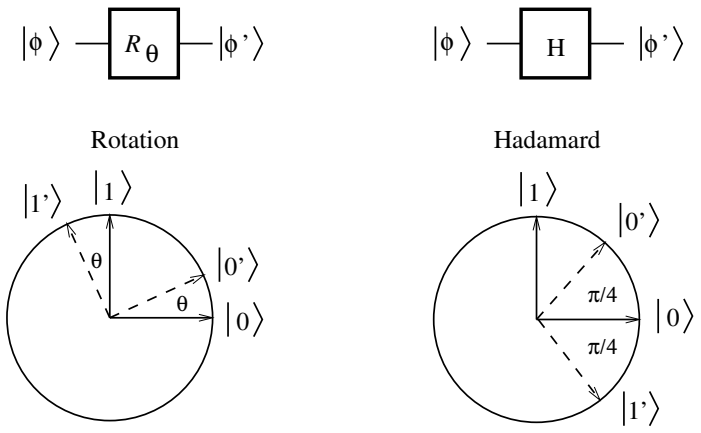

Figure 2.2: *Two single qubit quantum gates and their action on basis states.*

states by some angle θ:

$$\begin{aligned} |0\rangle &\mapsto |0'\rangle = \cos\theta\ |0\rangle + \sin\theta\ |1\rangle \\ |1\rangle &\mapsto |1'\rangle = -\sin\theta\ |0\rangle + \cos\theta\ |1\rangle\,, \end{aligned}$$

and the Hadamard gate effects the Fourier transform over $\mathcal{Z}_2$:

$$\begin{aligned} |0\rangle &\mapsto |0'\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\mapsto |1'\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

These are depicted in Figure 2.2. The set of all one-bit quantum gates and the two-bit controlled not gate together form a universal basis for quantum computation [10].

Some of the wires in the circuit are designated as input wires and some as output. Others form the workspace for the computation. Given a quantum circuit, the computation on an input $x \in \{0,1\}^n$ proceeds as follows. First, the input wires are initialised to $|0\rangle$ or $|1\rangle$ according to the bits of x, and the rest of the wires are set to $|0\rangle$. Next, the quantum gates are applied to the specified wires in sequence. Finally, the state of the output wires is observed by making a *measurement* in the standard (0/1) basis[1]. The measurement process has the following effect. If the final state of the wires is given by the superposition $|\phi\rangle = \sum_{y,z} \alpha_{y,z}\,|y,z\rangle$, where the y-part corresponds to the output wires, outcome y is observed

[1]Sometimes it is convenient to specify measurements in other bases. These correspond to first applying a unitary transformation (via a circuit) that effects a change of basis from the specified basis to the standard one, and then observing the state as before.

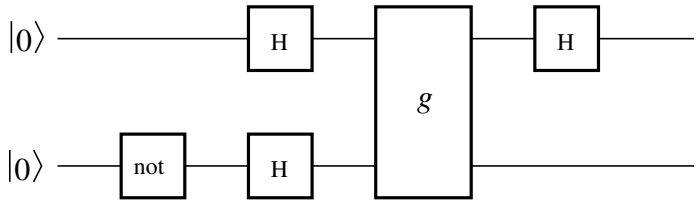

Figure 2.3: *A quantum oracle circuit that computes parity.*

with probability $p = \| \sum_z \alpha_{y,z} |y, z\rangle \|^2$, and the state of the wires "collapses" to the part of the superposition that is consistent with the observed value. In other words, the state of the wires becomes $\frac{1}{\sqrt{p}} \sum_z \alpha_{y,z} |y, z\rangle$, where $\sqrt{p}$ is the normalising factor. We say that a quantum circuit *computes* a function $f : \{0,1\}^n \to \{0,1\}^m$ if the probability that $f(x)$ is observed on input x is at least, say, $\frac{2}{3}$. We will use the terms *algorithm* and *circuit* interchangeably in this thesis.

In the black-box model, in addition to the usual quantum gates, we are provided with an *oracle gate*, which may act on more than a constant number of wires. We will be interested only in classical oracles, i.e., gates that compute classical functions of the type $g : \{0,1\}^\ell \to \{0,1\}^k$. Such gates act on $\ell + k$ wires. In a quantum oracle circuit, $\ell + k$ of the wires are designated as oracle wires, the first ℓ corresponding to the *query*, and the rest to the *reply* to the query. The oracle gate may only be applied to the oracle wires in a circuit. The action of the gate on basis states $\{|u, v\rangle\}$ is given by $|u, v\rangle \mapsto |u, g(u) \oplus v\rangle$, where $\oplus$ is the bit-wise exclusive-or operation on strings. Thus, if the last k of the oracle wires are all set to 0, the oracle replies with the value of g at the query point. Figure 2.3 shows an example of a quantum circuit with an oracle gate.

As explained earlier, oracle gates allow us to abstract out certain sets of operations (or an entire subroutine) that are often repeated in the process of computation. The parameter of interest in quantum oracle circuits is the number of oracle gates used to compute a function of interest. This will also be referred to as the number of queries (or *calls*) made to the oracle. The *query complexity* of computing a function given an oracle gate is the least number of such oracle gates required to compute the function using an oracle circuit.

We now describe a simple example to illustrate the concepts introduced above. Consider oracle gates for functions $g : \{0,1\} \to \{0,1\}$. Given such an oracle gate, the problem is to compute the parity (exclusive-or) of the two function values $g(0)$ and $g(1)$.

The circuit shown in Figure 2.3 computes this with no error using one oracle gate. To see this, we follow the evolution of the state of the wires as we apply the different gates and check that the output wire carries the desired result. Note that the state of the wires immediately before the oracle query is made is

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle).$$

The key observation is that the state $|b\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ (for $b = 0, 1$) is an eigenvector of the oracle gate with eigenvalue $(-1)^{g(b)}$. Thus, the state of the wires immediately after the oracle query is

$$\frac{1}{2}((-1)^{g(0)}|0\rangle + (-1)^{g(1)}|1\rangle)(|0\rangle - |1\rangle).$$

Applying the Hadamard gate to the first qubit results in

$$\frac{(-1)^{g(0)}}{\sqrt{2}}|g(0) \oplus g(1)\rangle (|0\rangle - |1\rangle),$$

which contains the required output in the first wire.

2.2 The two-party model

The two-party quantum communication model was introduced by Yao [78] and studied in (among others) [55, 31, 24, 9, 68]. Formally, a two-party communication protocol for a function is a partition of a quantum circuit for the function into two sets, where the input wires and the gates may be divided arbitrarily amongst the two, but all the output wires lie in one of the sets. The complexity of the protocol is the number of wires crossing between the two parts of the circuit.

In practice, it is more convenient to work with the following informal description. The model consists of two quantum players, say Alice and Bob, who wish to compute a joint function of the inputs supplied to them. The players follow a previously agreed *protocol* in order to compute the function. The protocol consists of a number of steps. In each step, one of four actions may occur: Alice may apply a unitary transformation to her set of qubits, Bob may apply a unitary transformation to his set, Alice may send some of her qubits to Bob, or vice-versa. At the end of all the steps, one player makes a measurement of her or his state to obtain the output. In cases where a specific player is required to know the answer, that player makes the measurement. The protocol is said to compute a function

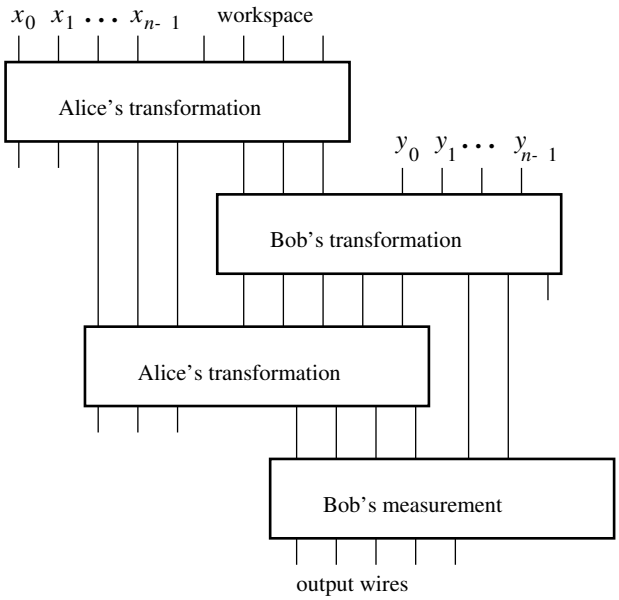

Figure 2.4: *A three-round quantum communication protocol.*

if the output is equal to the function value with "high" probability. We will mostly be interested in protocols that compute functions with probability at least $\frac{2}{3}$. The complexity of the protocol is the total number of quantum bits exchanged by the two players. The number of rounds taken by the protocol is the number of alternations in the exchange of bits between the players.

Note that there is no loss of generality in not allowing the players to measure a subset of their quantum bits in the intermediate steps of a protocol. This is because all measurements may be postponed to the end by the principle of safe storage [16]. Instead of making a measurement in some basis, the players may transform their state into that basis, copy the contents of the qubits to be measured on a fresh set of qubits (called the *ancilla*), and reverse the basis transformation.

The quantum communication complexity of a function is the complexity of a protocol that computes the function with the least number of quantum bits exchanged. It is possible to define other flavours of quantum communication complexity, depending on how much and what kind of error is allowed in the computation, but we will not delve into this here.

We will for the most part concentrate on communication in one round, since many

protocols of interest to us can be transformed to one-round protocols. These have a very simple structure. In such a protocol, Alice prepares some quantum bits in a superposition depending on her input and sends a subset of her qubits (the *encoding* of her input) to Bob, who measures it in a basis determined by his input. Below, we give an example of a one-round quantum communication protocol (or equivalently, a quantum encoding) for the equality function EQ to illustrate this.

The inputs to the function EQ are two n-bit strings x and y (given to Alice and Bob respectively), and the output is 1 if the two strings are equal, and 0 otherwise. Alice and Bob agree in advance on a binary error-correcting code $C \subset \{0,1\}^m$, of size at least 2^n, minimum distance at least d, and maximum distance at most $m-d$, for m and d to be specified later. Each n-bit string x is identified with a distinct codeword $\hat{x} \in C$. On input x, Alice prepares the state

$$|\phi_x\rangle \;=\; \frac{1}{\sqrt{m}} \sum_{i=0}^{m-1} (-1)^{\hat{x}_i} |i\rangle ,$$

and sends this encoding to Bob. On input y, Bob applies a unitary transformation U_y to $|\phi_x\rangle$ and measures it to check if the result is $|0\rangle$. He answers 1 if yes, and 0 otherwise. The only restriction on U_y is that it map $|\phi_y\rangle$ to $|0\rangle$.

We claim that the protocol above solves EQ with probability at least $\frac{2}{3}$ if m, d are chosen appropriately. If Alice and Bob have the same string x, then the result of Bob's measurement is always $|0\rangle$, and so the correct answer is obtained with probability one. If Bob has a string y different from Alice's string x, then the probability with which Bob gets $|0\rangle$ is

$$\begin{aligned} |\langle 0| U_y |\phi_x\rangle|^2 &= |\langle \phi_y| \phi_x\rangle|^2 \\ &= \left| \frac{1}{m} \sum_{i=0}^{m-1} (-1)^{\hat{x}_i + \hat{y}_i} \right|^2 \\ &= \left| 1 - 2\frac{d(\hat{x},\hat{y})}{m} \right|^2 , \end{aligned}$$

where $d(\hat{x},\hat{y})$ is the Hamming distance between the strings (i.e., number of bit positions i where $\hat{x}_i \neq \hat{y}_i$). If

$$\begin{aligned} \min_{x,y} d(\hat{x},\hat{y}) \geq d &\geq \frac{m}{2}(1 - \frac{1}{\sqrt{3}}), \quad \text{and} \\ \max_{x,y} d(\hat{x},\hat{y}) \leq m - d &\leq \frac{m}{2}(1 + \frac{1}{\sqrt{3}}), \end{aligned}$$

the above probability is at most $\frac{1}{3}$. Thus, Bob will get the correct answer with probability at least $\frac{2}{3}$.

It is well known (and can easily be shown by a counting argument) that for every $d < \frac{m}{2}$ there is a code of size at least $2^{m(1-H(\frac{d}{m}))-1}$ in $\{0,1\}^m$, with minimum distance at least d, and maximum distance at most $m-d$, where H is the binary entropy function. Thus, choosing d to be a suitable constant fraction of m, and $m = O(n)$ to be sufficiently large, we get an $O(\log n)$ quantum bit protocol (or encoding) for EQ.

2.3 One-way quantum finite automata

A one-way quantum finite automaton (QFA) is a theoretical model for a quantum computer with finite workspace. Models for such space-restricted quantum computers were first considered by [59, 54]. However, these models allow only a limited set of observations to be made during the computation process. The model we describe below rectifies this situation by allowing any *orthogonal* measurement as a valid intermediate computational step. Our model may be seen as a finite memory version of mixed state quantum computers (cf. [2]). Note that we do not allow the more general "positive operator valued measurements" [66] because the implementation of such measurements involves the joint unitary evolution of the state of the automaton with a fresh set of ancilla qubits, which runs against the (fixed finite workspace) spirit of the model.

In abstract terms, we may define an enhanced QFA as follows. It has a finite set of basis states Q, which consists of three parts: accepting states, rejecting states and non-halting states. The sets of accepting, rejecting and non-halting basis states are denoted by $Q_{\rm acc}, Q_{\rm rej}$ and $Q_{\rm non}$, respectively. One of the states, q_0, is distinguished as the starting state.

Inputs to a QFA are words over a finite alphabet Σ. We shall also use the symbols '¢' and '\$' that do not belong to Σ to denote the left and the right end-marker, respectively. The set $\Gamma = \Sigma \cup \{¢, \$\}$ denotes the working alphabet of the QFA. For each symbol $\sigma \in \Gamma$, a QFA has a corresponding "superoperator" $\mathcal{U}_\sigma$ which is given by a composition of a finite sequence of unitary transformations and orthogonal measurements on the space $\mathbb{C}^Q$. A QFA is thus defined by describing $Q, Q_{\rm acc}, Q_{\rm rej}, Q_{\rm non}, q_0, \Sigma$, and $\mathcal{U}_\sigma$ for all $\sigma \in \Gamma$.

At any time, the state of a QFA can be described by a density matrix with support

in $\mathbb{C}^Q$. The computation starts in the state $|q_0\rangle\langle q_0|$. Then transformations corresponding to the left end marker '¢,' the letters of the input word x and the right end marker '\$' are applied in succession to the state of the automaton, unless a transformation results in acceptance or rejection of the input. A transformation corresponding to a symbol $\sigma \in \Gamma$ consists of two steps:

1. First, $\mathcal{U}_\sigma$ is applied to $\boldsymbol{\rho}$, the current state of the automaton, to obtain the new state $\boldsymbol{\rho}'$.

2. Then, $\boldsymbol{\rho}'$ is measured with respect to the observable $E_{\text{acc}} \oplus E_{\text{rej}} \oplus E_{\text{non}}$, where $E_{\text{acc}} = \text{span}\{|q\rangle \mid q \in Q_{\text{acc}}\}$, $E_{\text{rej}} = \text{span}\{|q\rangle \mid q \in Q_{\text{rej}}\}$, $E_{\text{non}} = \text{span}\{|q\rangle \mid q \in Q_{\text{non}}\}$. The measurement has the following effect. The outcome E_i is observed with probability equal to $\text{Tr}(P_i\boldsymbol{\rho}')$, where P_i is the orthogonal projection onto E_i, and the state "collapses" to $P_i\boldsymbol{\rho}'P_i$ (suitably normalised). If we observe E_{acc} (or E_{rej}), the input is accepted (or rejected). Otherwise, the computation continues with the state $P_{\text{non}}\boldsymbol{\rho}'P_{\text{non}}$ (normalised), and the next transformation, if any, is applied.

We regard these two steps together as reading the symbol σ.

The model of QFA as defined in [54] differs from this model in that the superoperators $\mathcal{U}_\sigma$ are all required to be given by unitary transformations U_σ.

A QFA M is said to *accept* (or *recognize*) a language L with probability $p > \frac{1}{2}$ if it accepts every word in L with probability at least p, and rejects every word not in L with probability at least p.

A *reversible finite automaton* (RFA) is a QFA such that, for any $\sigma \in \Gamma$ and $q \in Q$, $\mathcal{U}_\sigma |q\rangle = |q'\rangle$ for some $q' \in Q$. In other words, the operator $\mathcal{U}_\sigma$ is a permutation over the basis states.

The *size* of a finite automaton is defined as the number of (basis) states in it. The "space used by the automaton" refers to the number of (qu)bits required to represent an arbitrary automaton state.

Chapter 3

Bounds for quantum communication

This chapter concentrates on results in quantum communication complexity. Most of these have previously been reported in [8] and [61].

The model of quantum communication is introduced in Section 2.2 of Chapter 2. Background material from information theory required for this chapter is summarised in Section A.1 of the appendix. We first discuss the classical information content of a quantum state in Section 3.1. We then turn to a study of quantum random access codes and their applications in Section 3.2. The technique developed in Section 3.2 has a novel application to computing, which we present in Section 3.3. We finish with a discussion of the chapter and open problems in the area of quantum communication.

3.1 An alternative to Holevo's theorem

Holevo's theorem [50], a fundamental result from quantum information theory, bounds the amount of classical information one can extract from a quantum encoding. If the encoding is over m qubits, the theorem says that it reveals no more than m bits of information about the message encoded. More precisely,

Theorem 3.1.1 (Holevo) *Let $x \mapsto \boldsymbol{\sigma}_x$ be any quantum encoding of bit strings into mixed quantum states, let X be a random variable over the strings with a distribution given by $\Pr[X = x] = p_x$, and let $\boldsymbol{\sigma} = \sum_x p_x \boldsymbol{\sigma}_x$ be the state corresponding to the encoding of*

the random variable X. If Y is any random variable obtained by performing a measurement on the encoding, then

$$I(X:Y) \quad \leq \quad S(\boldsymbol{\sigma}) - \sum_x p_x S(\boldsymbol{\sigma}_x).$$

Often we wish to bound not the mutual information $I(X:Y)$, but instead the probability of decoding X correctly from its encoding, i.e., $\Pr[Y = X]$, in terms of the number of qubits used in the encoding (see, e.g., [55, 8]). A bound on the probability of correct decoding may be obtained via Fano's inequality [34], which says that $\delta n - H(\delta) \leq I(X:Y)$, where $\delta = \Pr[Y = X]$, and X is a random variable over n-bit strings. Since Theorem 3.1.1 implies $I(X:Y) \leq S(\boldsymbol{\sigma}) \leq m$, where m is the number of qubits used in the encoding (cf. Fact A.1.1), we get $\delta n - H(\delta) \leq m$. This in turn may be used to derive a lower bound for m.

In this section, we present a simple alternative to this route by directly bounding the decoding probability achievable by any quantum encoding. Our bound thus obviates the need for a translation of in-probability statements into statements about mutual information in cases such as mentioned above.

Theorem 3.1.2 *Let X be a random variable over bit strings which are encoded as mixed states over m qubits and let $P(X,d)$ denote the net probability of the d most likely strings in the sample space of X. If Y is any random variable obtained by making some measurement of the encoding of X, then*

1. *there is a decoding procedure $\mathcal{D}_0$ such that*

$$\Pr[\mathcal{D}_0(Y) = X] \quad \geq \quad 2^{-H(X|Y)},$$

   where $H(X|Y)$ is the conditional Shannon entropy of X with respect to Y; and

2. *for any decoding function $\mathcal{D}$,*

$$\Pr[\mathcal{D}(Y) = X] \quad \leq \quad P(X, 2^m).$$

In particular, this implies that when X is distributed uniformly, the mutual information $I(X:Y)$ of X and Y is at most m. Thus, the bounds obtained from our theorem in typical applications are always at least as good as those derived from Theorem 3.1.1.

Specifically, for encodings of n-bit strings which can be decoded correctly with probability at least δ, we get $m \geq n - \log \frac{1}{\delta}$.

We illustrate how the above theorem can give us asymptotically sharper bounds on the number of qubits used in an encoding than an application of Holevo's theorem. Consider an encoding of n-bits into $n+1$ orthogonal states $|i\rangle$. Half the strings are encoded as $|0\rangle$, a fourth as $|1\rangle$, an eighth as $|2\rangle$, and so on. A random codeword from this code can be decoded with probability exactly $(n+1)2^{-n}$, which yields the correct answer for the number of qubits used by invoking our bound. On the other hand, the mutual information $I(X:Y)$ between the encoded string and its decoding is

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \cdots + \frac{n}{2^n} + \frac{n}{2^n},$$

which sums up to $2 - 2^{-(n-1)}$. This gives us a lower bound of at most 2 when combined with Holevo's theorem.

We now give a proof of Theorem 3.1.2. We first prove the lower bound on the decoding probability.

Consider random variables X and Y as in the statement of Theorem 3.1.2. We describe a natural decoding procedure $\mathcal{D}_0$ and then show that it satisfies the requirement of the theorem. On input y, the decoding algorithm outputs x such that $p_{x|y} = \max_{x'} \; p_{x'|y}$, where $p_{x|y} = \Pr[X = x | Y = y]$. Let $p_y^{\max}$ denote this probability and let x_y denote the corresponding x.

Claim 3.1.3 *The procedure $\mathcal{D}_0$ decodes correctly with probability at least $2^{-H(X|Y)}$.*

Proof: The probability of correct decoding is equal to

$$\begin{aligned} & \Pr[\mathcal{D}_0(Y) = X] \\ = \; & \sum_y \Pr[X = x_y | Y = y] \cdot \Pr[Y = y] \\ = \; & \mathrm{E}\left[p_Y^{\max}\right]. \end{aligned}$$

Now, $H(X|Y=y) = -\sum_x p_{x|y} \log p_{x|y} \geq -\log p_y^{\max}$. So $p_y^{\max} \geq 2^{-H(X|Y=y)}$. Taking expectation over Y, and noting that $2^{-(\cdot)}$ is a convex function, we have

$$\begin{aligned} \mathrm{E}\left[p_Y^{\max}\right] & \geq \mathrm{E}\left[2^{-H(X|Y=y)}\right] \\ & \geq 2^{-\mathrm{E}[H(X|Y=y)]} \\ & = 2^{-H(X|Y)}, \end{aligned}$$

which gives us the claimed lower bound on the decoding probability. ■

We now turn to the upper bound on the probability of correct decoding. Consider any encoding of strings $x \mapsto \{q_{x,i}, |\phi_{x,i}\rangle\}$ into mixed states over m qubits, and any decoding procedure $\mathcal{D}$. The output of $\mathcal{D}$ may be viewed as the outcome of a measurement given by orthogonal projections $\{P_x\}$ in the Hilbert space of the encoding augmented with some ancilla. The probability may then be bounded as

$$\begin{aligned}
&\Pr[\mathcal{D}(Y) = X] \\
&= \sum_x \Pr[\mathcal{D}(Y) = x] \cdot \Pr[X = x] \\
&= \sum_x p_x \sum_i q_{x,i} \| P_x |\phi_{x,i}\rangle \|^2 \\
&\leq \sum_x p_x \| P_x |\phi_x\rangle \|^2 , \qquad (3.1)
\end{aligned}$$

where $p_x = \Pr[X = x]$, and $|\phi_x\rangle$ is the pure state $|\phi_{x,i}\rangle$ that maximises the probability $\| P_x |\phi_{x,i}\rangle \|^2$ of obtaining the correct outcome x when its encoding is measured. (In all the expressions in this section, the ancilla qubits used in the measurement have been suppressed for ease of notation.) We can now bound the decoding probability by using the following claim.

Claim 3.1.4 $\sum_x \| P_x |\phi_x\rangle \|^2 \leq 2^m$.

Proof: Let E be the subspace spanned by the codewords $|\phi_x\rangle$, and let Q be the projection onto E. Since the codes are over m qubits, E has dimension at most 2^m. Let $\{|e_i\rangle\}$ be an orthonormal basis for E. Let $\{|\hat{e}_{x,j}\rangle\}$ be an orthonormal basis for the range of P_x. The union of all these bases $\{|\hat{e}_{x,j}\rangle\}$ is an orthonormal basis for the entire decoding Hilbert space. Now,

$$\begin{aligned}
\| P_x |\phi_x\rangle \|^2 &= \sum_j |\langle \hat{e}_{x,j} | \phi_x \rangle|^2 \\
&\leq \sum_j \| Q |\hat{e}_{x,j}\rangle \|^2 .
\end{aligned}$$

The last inequality follows because the length of the projection of any vector onto a space W is at least the length of its projection onto a subspace V of W. Observe that $\| Q |\hat{e}_{x,j}\rangle \|^2 = \sum_i |\langle e_i | \hat{e}_{x,j}\rangle|^2$. So,

$$\sum_x \| P_x |\phi_x\rangle \|^2 \leq \sum_i \sum_{x,j} |\langle e_i | \hat{e}_{x,j}\rangle|^2$$

$$\begin{aligned} &\leq \sum_i \| e_i \|^2 \\ &\leq 2^m, \end{aligned}$$

since the orthonormal basis $\{|e_i\rangle\}$ for E has size at most 2^m, which is a bound on the dimension of E. ■

By (3.1), the probability of correct decoding is at most $\sum_x p_x \| P_x |\phi_x\rangle \|^2$. From the claim above, this expression is equal to $\sum_x p_x \lambda_x$, where $0 \leq \lambda_x \leq 1$ and $\sum_x \lambda_x \leq 2^m$. The maximum over all such $\{\lambda_x\}$ of this quantity may easily be seen to be bounded by the sum of the 2^m largest probability masses p_x, i.e., by $P(X, 2^m)$. Moreover, for any given X and m, there is a natural pair of encoding and decoding functions that achieves this bound. This implies that the bound is tight.

Finally, we clarify how Theorem 3.1.2 may be applied in a communication complexity context (as opposed to a coding context) where more than one message may be exchanged by the communicating parties. The missing link is the following lemma due to Kremer [55] based on a technique of Yao [78].

Lemma 3.1.5 (Kremer) *Let P be an m-qubit communication protocol between two parties, Alice and Bob, such that Bob's actions are independent of his input. Then the state of Bob at the end of the protocol has support in a fixed 2^m dimensional subspace.*

In other words, while Bob's state may depend on Alice's input, it has support in a subspace that is independent of Alice's input. In such a case, we may always convert any multiple round protocol into a single round protocol that uses the same number of quantum bits, as in [9]. Alice and Bob agree on a common mapping between the standard basis over m qubits and the 2^m dimensional subspace prior to executing the protocol. Alice then simulates the entire original protocol herself, encodes Bob's part of the state into m qubits according to the agreed upon mapping, and sends it across to Bob. Bob can then retrieve the state in the original protocol by inverting the basis change. We thus end up with an m-qubit *coding* protocol to which our theorem may be applied.

3.2 Random access codes

In light of the limitation of quantum states in encoding classical information exposed in the previous section, it is tempting to conclude that the exponentially many degrees

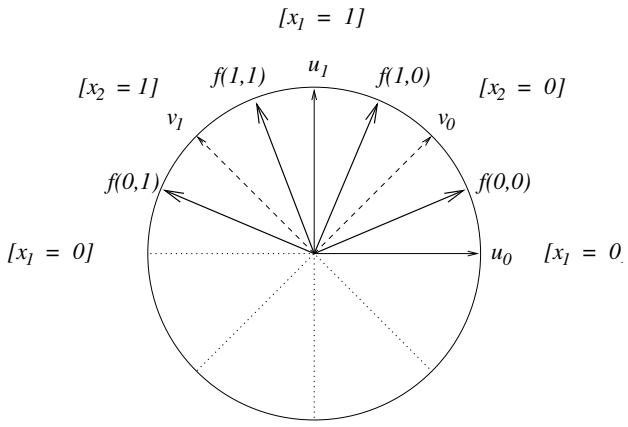

Figure 3.1: *A two-into-one quantum encoding with probability of success* ≈ 0.85.

of freedom latent in the description of a quantum system must necessarily stay hidden or inaccessible. However, the situation is more subtle, since in quantum mechanics, the recipient of a set of qubits has a choice of measurement he can make to extract information about their state. In general, these measurements do not commute. Thus making a particular measurement will, in general, disturb the system, thereby destroying some or all the information that would have been revealed by another possible measurement. This opens up the possibility of quantum *random access encoding*.

Definition 3.2.1 *An* (n, m, p)-random access encoding *is a function* f *that maps* n*-bit strings to mixed states over* m *qubits such that, for every* $i \in \{1, \ldots, n\}$*, there is a measurement* $\mathcal{O}_i$ *with outcome* 0 *or* 1 *that has the property that for all* $x \in \{0,1\}^n$*,*

$$\Pr\left[\mathcal{O}_i(f(x)) = x_i\right] \;\geq\; p.$$

Such quantum random access encoding with $n \gg m$ does not necessarily violate Holevo's bound: the recipient of the encoded string cannot make the measurements corresponding to the n different bits in succession to recover all the encoded bits with a good chance of success.

Indeed, this intuition is reinforced by the following encoding of two bits into one quantum bit.

Example: Let $|u_0\rangle = |0\rangle$, $|u_1\rangle = |1\rangle$, and $|v_0\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)$, $|v_1\rangle = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle)$. Define $f(x_1, x_2)$, the encoding of the string x_1x_2 to be $|u_{x_1}\rangle + |v_{x_2}\rangle$ normalised, unless $x_1x_2 =$

01, in which case it is $-|u_0\rangle + |v_1\rangle$ normalised (see Figure 3.1). The decoding functions are defined as follows: for the first bit x_1, we measure the message qubit according to the u basis and associate $|u_0\rangle$ with $x_1 = 0$ and $|u_1\rangle$ with $x_1 = 1$. Similarly, for the second bit, we measure according to the v basis, and associate $|v_0\rangle$ with $x_2 = 0$ and $|v_1\rangle$ with $x_2 = 1$. It is easy to verify that for all four codewords, and for any $i = 1, 2$, the angle between the codeword and the correct subspace is $\pi/8$. Hence the success probability is $\cos^2(\pi/8) \approx 0.85$. However, it is not possible to recover both the bits simultaneously with good probability. ■

A similar encoding of three bits into one qubit with success probability ≈ 0.78 is possible [28]. The next lemma shows that such classical encoding is not possible.

Lemma 3.2.1 *No classical $(2, 1, p)$-random access encoding exists for any $p > \frac{1}{2}$.*

Proof: Suppose for contradiction that there is a such a classical encoding. Let $f : \{0,1\}^2 \times R \mapsto \{0,1\}$ be the corresponding probabilistic encoding function and $V_i : \{0,1\} \times R' \mapsto \{0,1\}$ the probabilistic decoding functions. If we let y_i be the random variable $V_i(f(x,r), r')$, then for any $x \in \{0,1\}^2$, and any $i \in \{1,2\}$, $\Pr_{r,r'}(y_i = x_i) \geq p$.

We first give a geometric characterisation of the decoding functions. Each V_i clearly depends only on the encoding, which is either 0 or 1. Define the point P^j (for $j = 0, 1$) in the unit square $[0,1]^2$ as $P^j = (a_1^j, a_2^j)$, where $a_i^j = \Pr_{r'}(V_i(j, r') = 1)$. The point P^0 characterises the decoding functions when the encoding is 0, and P^1 characterises the decoding functions when the encoding is 1. For example, $P^1 = (1, 1)$ means that given the encoding 1, the decoding functions return $y_1 = 1$ and $y_2 = 1$ with certainty, and $P^0 = (0, \frac{1}{4})$ means that given the encoding 0, the decoding functions return $y_1 = 0$ and with probability $\frac{1}{4}$, $y_2 = 1$.

Now fix the decoding functions V_1, V_2. They define two points P^0 and P^1 in $[0,1]^2$. Given Bob's strategy and an input $x \in \{0,1\}^2$ Alice can choose (based on r) whether to encode x as 0 or 1. Let us say that Alice encodes x as 0 with probability p_x. Let us denote by $P^x = (a_1(x), a_2(x))$ the point with $a_i(x) = \Pr_{r,r'}(y_i = 1)$. As x is encoded as 0 with probability p_x and as 1 with probability $1 - p_x$ it follows that

$$P^x = p_x P^0 + (1 - p_x) P^1$$

Thus, for any x, P^x lies on the line segment connecting the two points P^0 and P^1. However, for the encoding to be a valid two-into-one encoding, the point P^x should lie *strictly* inside the quarter of the unit square $[0,1]^2$ closest to (x_1, x_2). The line connecting P^0 and P^1

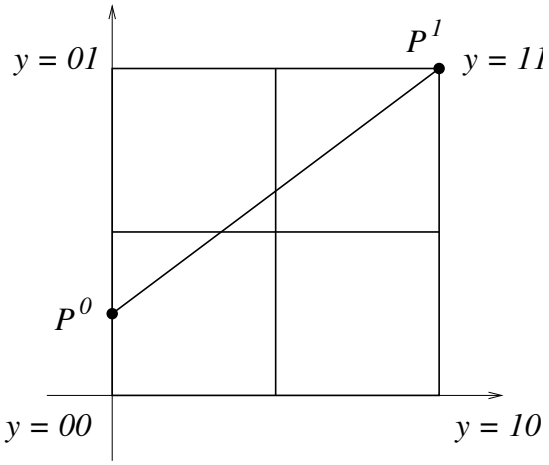

Figure 3.2: *A geometric characterisation of the probabilistic decoding functions of a two-into-one code.*

intersects the interiors of only three of the four quarters of the unit square $[0,1]^2$. For instance, if P^0 and P^1 are as in the example above, then the line connecting them does not pass through the lower right quarter (see Figure 3.2). Thus, for the string x_1x_2 which is favoured by that quarter (e.g., the string $x = 10$ in the example), either V_1 or V_2 errs with probability at least a half—a contradiction. ■

We can, in fact, give quite a precise characterisation of how much compression classical encoding can achieve asymptotically.

3.2.1 Bounds for classical codes

We first prove a lower bound on the number of bits required for classical random access encoding, and then show that there are classical codes that nearly achieve this bound.

Theorem 3.2.2 *Let $\frac{1}{2} < p \leq 1$. For any classical (n, m, p) random access encoding, $m \geq (1 - H(p))n$.*

Proof: Suppose there is such a (possibly probabilistic) encoding f. Let $X = X_1 \cdots X_n$ be chosen uniformly at random from $\{0,1\}^n$, and let $Y = f(X) \in \{0,1\}^m$ be the corresponding encoding. Let Z be the random variable with values in $\{0,1\}^n$ obtained by generating the bits $Z_1 \cdots Z_n$ from Y using the n decoding functions.

The mutual information of X and Y is clearly bounded by the number of bits in Y:

$$I(X:Y) \;\leq\; H(Y) \;\leq\; m.$$

We now that it is, in fact, bounded from below by $(1 - H(p))n$, thus getting our lower bound. We have

$$I(X:Y) \;=\; H(X) - H(X|Y) \;=\; n - H(X|Y).$$

Using standard properties of Shannon entropy, we get

$$H(X|Y) \;\leq\; H(X|Z) \;\leq\; \sum_{i=1}^{n} H(X_i|Z) \;\leq\; \sum_{i=1}^{n} H(X_i|Z_i).$$

Given that $\Pr[Z_i = X_i] \geq p$, it is not difficult to show (by using the concavity of entropy) that $H(X_i|Z_i) \leq H(p)$. It follows that $H(X|Y) \leq H(p)n$, and that $I(X:Y) \geq (1-H(p))n$, as we intended to show. ■

We now present a classical encoding scheme that asymptotically matches the lower bound derived above.

Theorem 3.2.3 *For any $p > \frac{1}{2}$ there is a classical (n, m, p)-random access encoding with*

$$m = (1 - H(p))n + O(\log n).$$

Proof: If $p > 1 - \frac{1}{n}$, $H(p) \leq \frac{\log n + 2}{n}$ and there is a trivial encoding—the identity map. So we turn to the case where $p \leq 1 - \frac{1}{n}$.

We use a code $S \subseteq \{0,1\}^n$ such that, for every $x \in \{0,1\}^n$, there is a $y \in S$ within Hamming distance $(1 - p - \frac{1}{n})n$. It is known (see, e.g., [32]) that there is such a code S of size

$$|S| \;=\; 2^{(1-H(p+\frac{1}{n}))n+2\log n} \leq 2^{(1-H(p))n+4\log n}.$$

Let $S(x)$ denote the codeword closest to x. One possibility is to encode a string x by $S(x)$. This would give us an encoding of the right size. Further, for every x, at least $(p+\frac{1}{n})n$ out of the n bits would be correct. This means that the probability (over all bits i) that $x_i = S(x)_i$ is at least $p + \frac{1}{n}$. However, for our encoding we need this probability to be at least p for *every* bit, not just on average over all bits. So we introduce the following modification.

Let r be an n-bit string, and π be a permutation of $\{1, \ldots, n\}$. For a string $x \in \{0,1\}^n$, let $\pi(x)$ denote the string $x_{\pi(1)}x_{\pi(2)} \cdots x_{\pi(n)}$.

Consider the encoding $S_{\pi,r}$ defined by $S_{\pi,r}(x) = \pi^{-1}(S(\pi(x+r))+r$. We show that if π and r are chosen uniformly at random, then for any x and any index i, the probability that the ith bit in the encoding is different from x_i is at most $1-p-\frac{1}{n}$. First, note that if i is also chosen uniformly at random, then this probability is clearly bounded by $1-p-\frac{1}{n}$. So all we need to do is to show that this probability is independent of i.

If π and r are uniformly random, then $\pi(x+r)$ is uniformly random as well. Furthermore, for a fixed $y = \pi(x+r)$, there is exactly one r corresponding to any permutation π that gives $y = \pi(x+r)$. Hence, if we condition on $y = \pi(x+r)$, all π (and, hence, all $\pi^{-1}(i)$) are equally likely. This means that the probability that $x_i \neq S_{\pi,r}(x)_i$ (or, equivalently, that $\pi(x+r)_{\pi^{-1}(i)} \neq (S(\pi(x+r))_{\pi^{-1}(i)})$ for random π and r is just the probability of $y_j \neq S(y)_j$ for random y and j. This is clearly independent of i (and x).

Finally, we show that there is a small set of permutation-string pairs such that the desired property continues to hold if we choose π, r uniformly at random from *this* set, rather than the entire space of permutations and strings. We employ the probabilistic method to prove the existence of such a small set of permutation-string pairs.

Let $\ell = n^3$, and let the strings $r_1, \ldots, r_\ell \in \{0,1\}^n$ and permutations $\pi_1, \ldots, \pi_\ell$ be chosen independently and uniformly at random. Fix $x \in \{0,1\}^n$ and $i \in [1..n]$. Let X_j be 1 if $x_i \neq S_{\pi_j,r_j}(x)_i$ and 0 otherwise. Then $\sum_{j=1}^{\ell} X_j$ is a sum of ℓ independent Bernoulli random variables, the mean of which is at most $(1-p-\frac{1}{n})\ell$. Note that $\frac{1}{\ell}\sum_{j=1}^{\ell} X_j$ is the probability of encoding the ith bit of x erroneously when the permutation-string pair is chosen uniformly at random from the set $\{(\pi_1, r_1), \ldots (\pi_\ell, r_\ell)\}$. By the Chernoff bound, the probability that the sum $\sum_{j=1}^{\ell} X_j$ is at least $(1-p-\frac{1}{n})\ell + n^2$ (i.e., that the error probability $\frac{1}{\ell}\sum_{j=1}^{\ell} X_j$ mentioned above is at least $1-p$) is bounded by $\mathrm{e}^{-2n^4/\ell} = \mathrm{e}^{-2n}$. Now, the union bound implies that the probability that the ith bit of x is encoded erroneously with probability more than $1-p$ for *any* x or i is at most $n2^n\mathrm{e}^{-2n} < 1$. Thus, there is a combination of strings $r_1, \ldots, r_\ell$ and permutations $\pi_1, \ldots, \pi_\ell$ with the property we seek. We fix such a set of ℓ strings and permutations.

We can now define our random access code as follows. To encode x, we select $j \in \{1, \ldots, \ell\}$ uniformly at random and compute $y = S_{\pi_j,r_j}(x)$. This is the encoding of x. To decode the ith bit, we just take y_i. For this scheme, we need $\log(\ell|S|) \leq \log \ell + \log|S| = (1-H(p))n + 7\log n$ bits. This completes the proof of the theorem. ∎

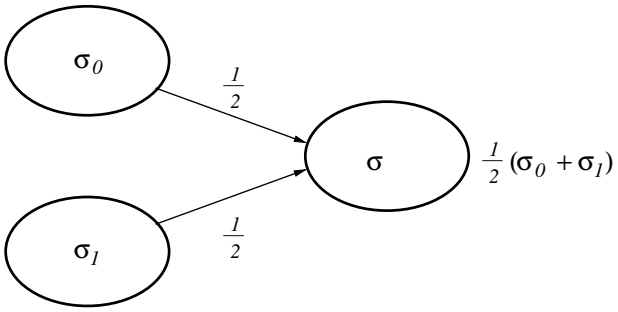

Figure 3.3: *Combining two distinguishable mixed states results in a state with higher entropy.*

3.2.2 Accumulation of information

In general, there is no a priori reason to rule out the existence of a (c^m, m, p)-encoding for constants $c > 1$, $p > \frac{1}{2}$ although this is not possible classically, for the reasons cited in Chapter 1. However, we showed in [8] that no more than a logarithmic factor compression is possible in the quantum case.

Theorem 3.2.4 *If a quantum (n, m, p) encoding exists with $p > \frac{1}{2}$ a constant, then $m \geq \Omega(\frac{n}{\log n})$.*

Thus, even though quantum random access encodings can beat classical encodings, they cannot be much more succinct.

Theorem 3.2.4 was based on amplification (by repetition of the code) which resulted in an encoding of *all* the original bits, as was shown by a hybrid argument originally due to [13] (and made explicit in [73]). Below, we derive a much stronger lower bound using very different, yet simpler techniques from quantum information theory.

The principle behind the lower bound may be described as that of "accumulation of information." Consider two mixed quantum states each with entropy at least s. Suppose we pick one of these with probability a half each. What is the entropy of the resulting mixed state? Clearly, if these mixed states were the same, the entropy would remain unchanged. On the other hand, if we could *distinguish between them* with success probability, say, 2/3 by making some measurement, then the two states differ substantially in at least a single bit position (in some basis). The mixture of the two states would then have approximately one bit more of randomness, and we expect the entropy of the system to increase accordingly. This is the essential content of our key lemma:

Lemma 3.2.5 *Let $\boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}_1$ be two density matrices, and let $\boldsymbol{\sigma} = \frac{1}{2}(\boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_1)$ be a random mixture of these matrices. If $\mathcal{O}$ is a measurement with outcome 0 or 1 such that making the measurement on $\boldsymbol{\sigma}_b$ yields the bit b with average probability p, then*

$$S(\boldsymbol{\sigma}) \quad \geq \quad \frac{1}{2}[S(\boldsymbol{\sigma}_0) + S(\boldsymbol{\sigma}_1)] + (1 - H(p)).$$

This lemma is a simple corollary of the classic Holevo theorem [50] stated as Theorem 3.1.1 above.

Proof of Lemma 3.2.5: Consider $\boldsymbol{\sigma}_b$ to be an encoding of the bit b. If X is an unbiased boolean random variable, then $\boldsymbol{\sigma}$ represents the encoding of X. Let Y be the outcome of the measurement of this encoding according to $\mathcal{O}$. By the hypothesis of the lemma, $\Pr[Y = X] = p$. It is easy to see from the concavity of the entropy function that

$$I(X:Y) \quad \geq \quad 1 - H(p)$$

(cf. Fano's inequality [34]). The lemma now follows from Theorem 3.1.1. ■

This principle of accumulation of information is applicable in other situations as well, as we will later see in Section 3.3.

3.2.3 The quantum lower bound

The principle presented in the previous section yields a bound for quantum random access codes that matches the *classical* upper bound of $(1 - H(p))n + O(\log n)$ shown in Section 3.2.1 up to the logarithmic additive term.

Theorem 3.2.6 *Any quantum (n, m, p)-random access encoding has $m \geq (1 - H(p))n$.*

We now prove this theorem. Consider any random access encoding with parameters n, m, p. Let $\boldsymbol{\rho}_x$ denote the density matrix corresponding to the encoding of the n-bit string x. The density matrix of a random codeword is given by $\boldsymbol{\rho} = \frac{1}{2^n}\sum_x \boldsymbol{\rho}_x$. We can bound the entropy of $\boldsymbol{\rho}$ by m by Fact A.1.1. Using Lemma 3.2.5, we can also prove a lower bound for the entropy of $\boldsymbol{\rho}$, and hence obtain a lower bound on m.

For any $y \in \{0,1\}^k$, where $0 \leq k \leq n$, let

$$\boldsymbol{\rho}_y \quad = \quad \frac{1}{2^{n-k}} \sum_{z \in \{0,1\}^{n-k}} \boldsymbol{\rho}_{zy}.$$

We claim that

Claim 3.2.7 $S(\boldsymbol{\rho}_y) \geq (1 - H(p))(n - k)$.

Proof: The proof is by downward induction on k. The base case $k = n$ is satisfied easily: $S(\boldsymbol{\rho}_y) \geq 0$ for all n-bit strings y.

Suppose the claim is true for $k + 1$. We have

$$\boldsymbol{\rho}_y \;=\; \frac{1}{2}(\boldsymbol{\rho}_{0y} + \boldsymbol{\rho}_{1y}).$$

By hypothesis,

$$S(\boldsymbol{\rho}_{by}) \;\geq\; (1 - H(p))(n - k - 1),$$

for $b = 0, 1$. Moreover, since the two density matrices are mixtures arising from strings that differ in the $(n - k)$th bit, the measurement $\mathcal{O}_{n-k}$ distinguishes them correctly with probability p. Thus, by Lemma 3.2.5, we get

$$S(\boldsymbol{\rho}_y) \;\geq\; \frac{1}{2}(S(\boldsymbol{\rho}_{0y}) + S(\boldsymbol{\rho}_{1y})) + (1 - H(p)),$$

which gives us the claimed bound. ■

Theorem 3.2.6 now follows by combining the claim (with y chosen to be the empty string) and the upper bound of m on the entropy. Observe that we could allow the measurement $\mathcal{O}_i$ to depend on the subsequent bits of the encoded string without affecting the lower bound. This means that the bound holds for *serial codes*, a type of code we introduced in [8], as well. Lower bounds for these codes imply space lower bounds for quantum finite automata. Rather than elaborating on this further, we will later present a slightly different proof for the space lower bound based on the same principle.

3.2.4 The effect of shared entanglement

Quantum communication with shared entanglement is an analogue of public coin randomised communication. In this model, the two communicating players are allowed to start with an arbitrarily large shared quantum state (independent of their inputs) at no cost to the protocol. Shared entanglement enables us to transfer information in highly non-intuitive ways, as demonstrated by the "teleportation" technique of [14].

Next, we address the question as to whether shared entanglement helps reduce the amount of communication required to construct random access codes. The situation here is that Alice is given the classical string to be encoded and Bob wishes to acquire a random access encoding of it. Can they accomplish this by exchanging fewer quantum bits

if they are allowed to set up an arbitrary (but fixed) shared state before the start of the protocol? The superdense coding technique of [15] can be used to halve the communication requirement given by the classical encoding scheme presented in Section 3.2.1. We show that this is essentially all that can be done by exploiting shared entanglement.

Theorem 3.2.8 *Any protocol for random access encoding of n bits with success probability p has complexity at least $(1 - H(p))n$. Moreover, Alice sends at least $\frac{1}{2}(1 - H(p))n$ quantum bits to Bob in the protocol.*

To see that this implies our claim, notice that Bob may do all the communication required to set up the initial shared state.

The proof of Theorem 3.2.8 is based on a different view of the principle presented in Section 3.2.2. It makes use of properties of von Neumann entropy more involved than those presented in Section A.1. These properties are discussed in detail in [66]. The first part of the theorem is relatively simple. We consider the protocol on a random input to Alice, and show that the entropy of Bob's state increases only when there is some communication between the two players, and it increases only by one per qubit exchanged. The lower bound for random access codes from the previous section now implies the bound. For the second part, we show that the mutual information between the input and Bob's state may increase only when Alice sends him some of her qubits. This increase may be at most two per qubit sent by Alice. (This part of the argument is identical to one used by Cleve, van Dam, Nielsen and Tapp [31].) On the other hand, since Bob's state can be used to extract an arbitrary bit of the input, we can show that it encodes a lot of information about the input.

Proof of Theorem 3.2.8: Consider the protocol on a random n-bit input to Alice. Let B^k be Bob's state after k qubits have been exchanged during the protocol ($k \geq 0$). We claim by induction that $S(B^k) \leq k$. Initially, $S(B^0) = 0$. This entropy remains unchanged if either Alice or Bob applies some unitary transformation to her/his quantum bits. If Alice sends Bob qubit Q as the kth qubit in the protocol, then $S(B^k) = S(QB^{k-1}) \leq S(Q) + S(B^{k-1}) \leq 1 + (k-1)$ by the subadditivity property of entropy. If Bob sends Alice the qubit, then $B^{k-1} = QB^k$, and by the Araki-Lieb inequality we have $S(B^{k-1}) \geq S(B^k) - S(Q) \geq S(B^k) - 1$, so $S(B^k) \leq k$. Since Bob's state is a random access encoding of Alice's input, it has entropy at least $(1 - H(p))n$ at the end of the protocol. This proves the first part of the theorem.

The second part of the theorem is a little more involved. Let B be the state of Bob at the end of the protocol on random input $X = X_1 \cdots X_n$, and B_x the state on a particular input x. Similarly, let B^k, B_x^k now be defined as the state of Bob after Alice has sent k qubits to him.

We first prove a lower bound of $(1 - H(p))n$ for $I(X : B)$. By Equation (A.1), we have

$$\begin{aligned} I(B : X_1 \cdots X_n) &= I(B : X_n) + I(BX_n : X_1 \cdots X_{n-1}) - I(X_n : X_1 \cdots X_{n-1}) \\ &= I(B : X_n) + I(BX_n : X_1 \cdots X_{n-1}). \end{aligned}$$

The second equality follows because all the X_i are independent classical random variables (hence $I(X_n : X_1 \cdots X_{n-1}) = 0$). Continuing this way, and applying inequality (A.2) we get

$$\begin{aligned} I(B : X) &= \sum_{i=1}^{n} I(BX_{i+1} \cdots X_n : X_1 \cdots X_i) \\ &\geq \sum_{i=1}^{n} I(BX_{i+1} \cdots X_n : X_i). \end{aligned}$$

Now, given $BX_{i+1} \cdots X_n$, we can use the measurement $\mathcal{O}_i$ for the random access code to construct a random variable Z_i such that $\Pr[Z_i = X_i] \geq p$. Thus, we have

$$I(BX_{i+1} \cdots X_n : X_i) \;=\; I(BX_{i+1} \cdots X_n Z_i : X_i) \;\geq\; I(Z_i : X_i).$$

As before, $I(Z_i : X_i) \geq 1 - H(p)$. Putting it all together we get

$$\begin{aligned} I(B : X) &\geq \sum_{i=1}^{n} I(BX_{i+1} \ldots X_n : X_i) \\ &\geq \sum_{i=1}^{n} I(Z_i : X_i) \\ &\geq (1 - H(p))n. \end{aligned}$$

Furthermore, since $S(XB) = S(X) + 2^{-n} \sum_x S(B_x)$, we have

$$I(X : B) \;=\; S(B) - 2^{-n} \sum_x S(B_x).$$

The quantity on the right is the *accessible information* $\chi(B)$ in B. To bound this, we show by induction that $\chi(B^k) \leq 2k$. Initially, $\chi(B^0) = 0$. This quantity remains unchanged when Alice and Bob do some computation on their parts of the system. By the

monotonicity property of accessible information, it only decreases when Bob sends Alice any of his quantum bits. When Alice sends Bob her kth qubit Q (Q_x on a particular input x), we have, as before, $S(B^k) = S(QB^{k-1}) \leq S(Q) + S(B^{k-1}) = 1 + S(B^{k-1})$, and $S(Q_x B_x^{k-1}) \geq S(B_x^{k-1}) - S(Q_x) = S(B_x^{k-1}) - 1$. So $\chi(B^k) \leq \chi(B^{k-1}) + 2 \leq 2k$. Combining this with the lower bound on the mutual information obtained above, we get the claimed result. ■

3.2.5 Rounds in communication complexity

Recently, Buhrman and de Wolf [26] observed that any one-round quantum communication protocol for the disjointness function (DISJ) gives rise to random access codes, and thus our lower bound applies there as well. For completeness, we include the argument here. In the problem DISJ, the two parties are given a subset of $[n] = \{0, 1, \ldots, n-1\}$ each. The problem is to determine whether the subsets are disjoint or not. If we represent the two subsets by $x, y \in \{0,1\}^n$, their characteristic vectors, we may express the function as $\mathrm{DISJ}(x, y) = \vee_i (x_i \wedge y_i)$. We may assume, w.l.o.g., that Alice sends the first message (based on her input x). Since at this point Bob should be able to infer $\mathrm{DISJ}(x, y)$ for any input y, he may chose y to corresponding to any singleton set $\{i\}$, and thus infer any bit of x he wishes to learn. Alice's message therefore defines a random access encoding, and has length $\Omega(n)$ for any bounded-error protocol. On the other hand, there is an $O(\sqrt{n})$-round protocol for DISJ that uses $O(\sqrt{n} \log n)$ quantum bits, as shown by Buhrman, Cleve and Wigderson [24]. Thus, by allowing more exchange of messages, the number of quantum bits transmitted to solve a communication task may be reduced substantially.

We show next that our results imply a much stronger exponential separation between quantum protocols differing in the number of rounds of communication. More specifically, we show that a natural problem has one-round quantum communication complexity $\Omega(n/\log n)$, whereas there is a *classical* two-round protocol for it with complexity $O(\log n)$.

We now define the communication problem mentioned above. Let $m \geq 1$ be an integer. Alice, the first player in the communication game, gets as input an element $i \in [m] = \{0, \ldots, m-1\}$ and a subset $M \subset [m]$. Bob, the second player, gets a function $f : [m] \to [m]$. The problem is to determine if $f(i) \in M$. We refer to this problem as Π_2 in the sequel.

Note that n, the problem size, is $\Theta(m \log m)$. The two-round $O(\log n)$ classical protocol for Π_2 is straightforward: Alice sends Bob the element i and Bob returns Alice the element $f(i)$ it is mapped to by the function f. Alice then knows the answer to the problem, which can be relayed to Bob. The protocol involves $O(\log m) = O(\log n)$ bits of communication. It is perhaps worth mentioning that the quantum communication complexity of Π_2 is $\Omega(\log n)$. This follows from the proof of the $\Omega(\log n)$ lower bound for DISJ given by Buhrman and de Wolf [26, Proposition 1]. They show that a special case of DISJ—when Alice has an arbitrary subset of $[n]$ and Bob has an arbitrary *singleton* subset of $[n]$—has quantum communication complexity $\Omega(\log n)$. This is clearly equivalent to the restriction of our problem when the input i to Alice is fixed (to, say, 0).

We now turn to one-round protocols for Π_2. First, note that the problem may be solved with $O(m)$ bits of classical communication in one round: Alice can send both i and M to Bob. Since the subset M may be represented as an m-bit vector, we get the abovementioned bound. We claim that this is the best possible (up to a constant factor) even when quantum communication is allowed. This quantum lower bound for Π_2 is proven by showing that any k-qubit protocol for it with probability $p > \frac{1}{2}$ of success results in an (m, k, p)-random access encoding.

Theorem 3.2.9 *Any one-round quantum protocol for* Π_2 *uses at least* $\Omega(m) = \Omega(n/\log n)$ *qubits of communication.*

Proof: A one-round k-qubit protocol for Π_2 in which Alice sends the first message defines a (m, k, p)-random access code as follows. We consider messages sent by Alice when i is fixed to be 0. Every m-bit string x is identified in the natural way with a subset $M \subseteq [m]$. The encoding of x is then defined as the message sent by Alice on input i, M. To recover the jth bit of x for an arbitrary $j \in [m]$, Bob does the measurement given by the one-round protocol on any input function f such that $f(0) = j$. By Theorem 3.2.6, $k = \Omega(m)$.

A similar bound holds for protocols in which Bob initiates the communication. Consider inputs to Π_2 where the input subset M is fixed to $\{1\}$, and the functions f are such that $f(j) \in \{0, 1\}$ for all j. Each binary string of length m may be identified with one such function f. The messages of Bob on the 2^m different strings define an (m, k, p)-random access encoding: to recover an arbitrary bit x_j of an encoded string, Alice may use the measurement defined by the protocol on input $i = j$ and $M = \{1\}$. Thus, we again have $k = \Omega(m)$. ■

Thus, the one-round quantum communication complexity is exponentially larger than its optimal (two-round) classical complexity.

3.3 Implications for finite automata

A *quantum finite automaton* (QFA) as defined by Kondacs and Watrous [54] differs from a deterministic finite automaton (DFA) in that its state is in general a superposition of the classical (basis) states. It starts in such a state, and when a new input symbol σ is seen, a corresponding unitary operator U_σ is applied to it. The state is then measured to check for acceptance, rejection or continuation. If the result of the measurement is 'continue,' the next symbol is read, otherwise the input is accepted or rejected. A QFA recognises a language if all the strings in it (or not in it) are accepted (respectively, rejected) with constant probability bounded away from $1/2$. In [8], we gave an exponential size lower bound for QFAs for checking if the input is a small even number, which is an almost immediate consequence of our lower bound for random access codes.

In this thesis, we study more general QFAs, in which instead of only applying a unitary transformation when a new input symbol is seen, we allow a combination of unitary operators and orthogonal measurements. (We refer the reader to Section 2.3 of Chapter 2 for a more formal definition of QFAs.) In the case of more general models such as quantum Turing machines such intermediate measurements do not increase the power of the model, since measurements can always be replaced by safe storage. However, in the case of QFAs, the space limitations inherent in the definition preclude the possibility of similar reasoning. Moreover, in this new model, the evolution of the system is no longer reversible, so the intuition from [54, 8] no longer applies. Indeed, this new model of QFA was suggested by Dorit Aharonov as a more physically appropriate model that might not suffer from unnecessary handicaps resulting from the reversibility property embedded in the definitions from [59, 54]. However, it is not hard to verify (by applying a technique of [67], also used in [54]) that such QFAs accept only regular languages. Moreover, we show that a stronger version of the bound of [8] continues to hold.

Theorem 3.3.1 *Let L_n be the language $\{w0 \mid w \in \{0,1\}^*,\ |w| \leq n\}$. Then,*

1. *L_n is recognised by a DFA of size $O(n)$,*
2. *L_n is recognised by some QFA, and*

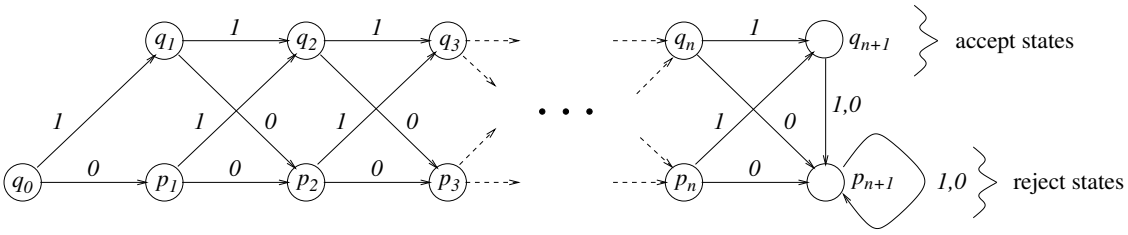

Figure 3.4: *A DFA that accepts the language* $L_n = \{w0 \mid w \in \{0,1\}^*, |w| \leq n\}$.

3. Any QFA recognising L_n *with some constant probability greater than* $\frac{1}{2}$ *has* $2^{\Omega(n)}$ *states.*

A $2^{\Omega(n)}$ versus $O(n)$ separation is the worst possible if only finite languages are considered: it is not hard to see that a finite language with largest word-length n is accepted by a *reversible* finite automaton of size $2^{O(n)}$. Moreover, any DFA for it has size at least n, for otherwise, by the Pumping Lemma, the DFA would accept infinitely many words. We also point out that the proof of this theorem implies that QFAs accept only a strict subset of the regular languages.

We now give the proof of Theorem 3.3.1, which compares classical and quantum automata for checking if a given input is a small even number (an even number less than 2^{n+1}). The first two parts of Theorem 3.3.1 are easy. Figure 3.4 shows a DFA with $2n+3$ states for the language L_n. Part 3 of the theorem may be shown to follow from the lower bound argument we give for random access codes (as we show in [8]) if only unitary operations are allowed during the computation. However, we will take a slightly different path based on the "principle of accumulation of information" (Lemma 3.2.5) that applies to general QFA as well.

Consider the evolution of the a quantum system under a random sequence of unitary transformations (V_i), where each V_i is either U_0 or U_1 (see Figure 3.5). Now suppose that the transformations U_0 and U_1 are distinguishable in the sense that for every superposition $|\phi\rangle$ of the system, $U_0 |\phi\rangle$ can be distinguished from $U_1 |\phi\rangle$ with success probability, say, $p > \frac{1}{2}$ by some fixed measurement. At each step, the system gains some information about the transformation applied to it, and we expect the entropy of the system to increase accordingly. In general, we could apply one of two arbitrary but distinguishable quantum operations on the system, and we would expect the same increase in entropy. This is exactly what Lemma 3.2.5 formalises.

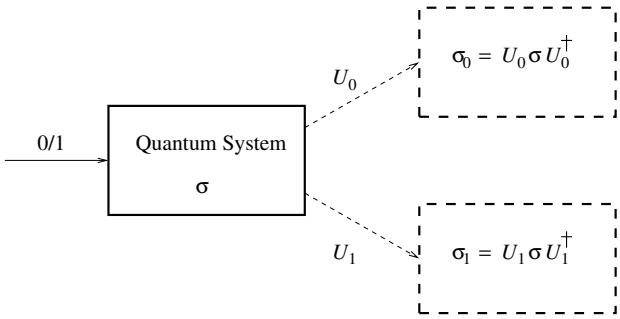

Figure 3.5: *A stream of random bits determining the evolution of a quantum system.*

We now prove Theorem 3.3.1 using this framework. We need the following definition first.

Definition 3.3.1 *An* r-restricted *one-way QFA for a language* L *is a one-way QFA that recognises the language with probability* $p > \frac{1}{2}$*, and which halts with non-zero probability before seeing the right end-marker only after it has read* r *letters of the input.*

We begin with a bound of $2^{(1-H(p))n}$ for the number of basis states in any n-restricted QFA M for L_n that has success probability p. Note that the evolution of M on reading stream of random bits corresponds exactly to that of the quantum system described above. We prove that, at the end of reading a random n-bit string, the state of M has entropy at least $(1 - H(p))n$. However, this entropy is bounded by $\log |Q|$ by Fact A.1.1, where Q is the set of basis states of M. This gives us the above bound.

Let $\boldsymbol{\rho}_k$ be the state of the QFA M after the kth symbol of a random n-bit input has been read ($0 \leq k \leq n$).

Claim 3.3.2 $S(\boldsymbol{\rho}_k) \geq (1 - H(p))k$.

Proof: Let $\mathcal{U}_\sigma$ be the superoperator of M corresponding to the symbol σ. Let E_0 be the span of the accepting basis states of M and let E_1 be the subspace orthogonal to it. Define the measurement $\mathcal{O}$ as applying the transformation $\mathcal{U}_\$$ (recall that '\$' is the right end-marker) and then measuring with respect to the observable $E_0 \oplus E_1$. We can now prove the claim by induction.

For $k = 0$, $S(\boldsymbol{\rho}_0) \geq 0$. Now assume that $S(\boldsymbol{\rho}_{k-1}) \geq (1 - H(p))(k - 1)$. After

the kth random input symbol is read, the state of M becomes

$$\boldsymbol{\rho}_k \;=\; \frac{1}{2}(\mathcal{U}_0(\boldsymbol{\rho}_{k-1}) + \mathcal{U}_1(\boldsymbol{\rho}_{k-1})).$$

By the definition of M, measuring $\mathcal{U}_b(\boldsymbol{\rho}_{k-1})$ according to $\mathcal{O}$ yields b with probability at least $p > \frac{1}{2}$. So by Lemma 3.2.5, we have

$$S(\boldsymbol{\rho}_k) \;\geq\; \frac{1}{2}\sum_{b=0,1} S(\mathcal{U}_b(\boldsymbol{\rho}_{k-1})) + (1 - H(p)). \tag{3.2}$$

But the entropy of a mixed state is preserved by unitary transformations (Fact A.1.2) and only increases on orthogonal measurement (Fact A.1.3), so

$$S(\mathcal{U}_b(\boldsymbol{\rho}_{k-1})) \;\geq\; S(\boldsymbol{\rho}_{k-1}) \geq (1 - H(p))(k-1).$$

Inequality (3.2) now gives us the claimed bound. ■

It only remains to show that the lower bound on the size of restricted QFA obtained above implies a lower bound on the size of general (i.e., unrestricted) QFA accepting L_n. We do this by showing that we can convert *any* one-way QFA to an r-restricted one-way QFA which is only $O(r)$ times as large as the original QFA. It follows that the $2^{\Omega(n)}$ lower bound on number of states of n-restricted QFA recognising L_n continues to hold for all QFA for L_n, exactly as stated in Theorem 3.3.1.

The idea behind the construction of a restricted QFA, given any QFA, is to carry the halting parts of the state of the original automaton as "distinguished" non-halting parts of the state of the new automaton till at least r more symbols of the input have been read since the halting part was generated or until the right end marker is encountered, and then mapping them to accepting or rejecting subspaces appropriately.

Lemma 3.3.3 *Let M be a QFA with $|M|$ states recognising a language L with probability p. Then there is an r-restricted QFA M' with $O(r\,|M|)$ states that recognises L with probability p.*

Proof: Let M be a QFA with Q as the set of basis states, Q_{acc} as the set of accepting states, Q_{rej} as the set of rejecting states, and q_0 as the starting state. Let M' be the automaton with basis state set

$$Q \;\cup\; (Q_{\text{acc}} \times \{0, 1, \ldots, r+1\} \times \{\text{acc}, \text{non}\}) \;\cup$$

$$(Q_{\text{rej}} \times \{0, 1, \ldots, r+1\} \times \{\text{rej}, \text{non}\}).$$

Let $Q_{\text{acc}} \cup (Q_{\text{acc}} \times \{0, 1, \ldots, r+1\} \times \{\text{acc}\})$ be its set of accepting states, let $Q_{\text{rej}} \cup (Q_{\text{rej}} \times \{0, 1, \ldots, r+1\} \times \{\text{rej}\})$ be the set of rejecting states, and let q_0 be the starting state.

The super-operators (which consist of a finite sequence of unitary operations and orthogonal measurements) for the new QFA are constructed as follows. All transitions but those in the last unitary transformation corresponding to any symbol are retained as before. The last unitary transformation is modified as below. If for a state $q \in Q$, there is a transition

$$|q\rangle \quad \mapsto \quad \sum_{q'} \alpha_{q'} |q'\rangle$$

in M on symbol σ, then in M', we have the following transitions. On the '\$' symbol, we have the same transition, and on $\sigma \neq \$$, we have

$$|q\rangle \quad \mapsto \quad \sum_{q' \notin Q_{\text{acc}} \cup Q_{\text{rej}}} \alpha_{q'} |q'\rangle + \sum_{q' \in Q_{\text{acc}} \cup Q_{\text{rej}}} \alpha_{q'} |q', 0, \text{non}\rangle .$$

The transitions from the states not originally in M are given by the following rules. For any symbol $\sigma \neq \$$, we have

$$|q, i, \text{non}\rangle \quad \mapsto \quad \begin{cases} |q, i+1, \text{non}\rangle & \text{if } i < r \\ |q, i+1, \text{acc}\rangle & \text{if } q \in Q_{\text{acc}} \text{ and } i = r \\ |q, i+1, \text{rej}\rangle & \text{if } q \in Q_{\text{rej}} \text{ and } i = r \end{cases}$$

On the '\$' symbol we have:

$$|q, i, \text{non}\rangle \quad \mapsto \quad \begin{cases} |q, i, \text{acc}\rangle & \text{if } q \in Q_{\text{acc}} \text{ and } i \leq r \\ |q, i, \text{rej}\rangle & \text{if } q \in Q_{\text{rej}} \text{ and } i \leq r \end{cases}$$

The rest of the transitions may be defined arbitrarily, subject to the condition of unitarity.

All measurements made in M are augmented with a complete measurement of the new states.

It is not difficult to verify that M' is an r-restricted one-way QFA (of size $O(r\,|M|)$) accepting the same language as M, and with the same probability. ■

As a simple consequence of the size lower bound derived above, we obtain:

Theorem 3.3.4 *The regular language* $\{0,1\}^*0$ *cannot be accepted by any QFA with probability bounded away from* $\frac{1}{2}$.

To see this, we note that any QFA that supposedly recognises this language also correctly recognises all words of length at most n of the language L_n, for every n. The proof of Theorem 3.3.1 now tells us that the number of states in the QFA is $2^{\Omega(n)}$ for every n, which is a contradiction.

3.4 Concluding remarks

Holevo's theorem is a recurring tool in shedding light on the limitations of quantum communication. While we have given a greatly simplified explanation for it that works well in many situations, our understanding of it is still incomplete. Apart from the case of random access codes, where we used the theorem in its full generality, there are other situations where it has found applications. One such is in proving communication bounds in the presence of shared entanglement [31]. Our result does not seem to apply to this case either. It would be useful to have an explanation of Holevo's theorem that extends to these cases as well.

Several open problems emerge from our study of random access codes. The first relates to private information retrieval (see, e.g., [27]). Theorem 3.2.8 implies a linear communication lower bound for the problem of information-theoretically secure private information retrieval with one database when constant probability of error is allowed. It is known that by increasing the number of databases to two, one can reduce the communication required to $O(\sqrt[3]{n})$, even for zero-error retrieval (see [52] for the latest developments in this field). It might be possible to reduce this even further using quantum communication.

Another problem relates to the relative power of communication using a different number of rounds. Generalisations of the problem Π_2 of Section 3.2.5 are known to witness exponential separation between k and $k+1$ rounds of communication ($k \geq 1$) in classical complexity theory [56]. We believe that our techniques can be extended to prove the same for quantum communication as well, and are currently investigating this.

Although information theoretic arguments prove to be useful in many contexts, we do not know of a way to use them to get interesting bounds for, say, the set disjointness problem discussed in Section 3.2.5. While [24] gives a general method for designing efficient quantum communication protocols, no such general paradigm for showing the limitations of

quantum communication is known. An approach that may be of relevance in this context is one recently introduced in [26]—that of proving bounds via polynomials.

Our study of quantum finite automata further motivates the study of restrictions on quantum computation forced by experimental considerations. Space requirements are likely to be a dominant consideration in the realisation of quantum computers because of the difficulties involved in maintaining entanglement across a large quantum system for extended periods of time. The study of space bounded quantum computation with more general space bounds thus assumes importance. It was recently proved [76] that quantum computation offers limited savings in space in universal models of computation (in contrast to the case of finite automata). A space S quantum computation can be accomplished in deterministic space $O(S^2)$, while deterministic computations using space S can be simulated on a quantum computer with only a constant factor increase in space. Some questions that remain unresolved are whether there are more time-efficient quantum simulations of classical processes that do not incur any (significant) space overhead, and whether bounded-error quantum computations can be carried out by bounded-error probabilistic machines with sub-quadratic blow-up in space. Answers to such questions will help us to better understand the power of space bounded quantum devices.

Chapter 4

Bounds for quantum computation

This chapter focuses on showing optimal (or nearly optimal) bounds for solving some problems in the quantum black-box model. Most of these bounds have previously been reported in [62].

Th black-box model is precisely defined in Section 2.1 of Chapter 2. We start with the problem of approximating the median in Section 4.1, and illustrate the technique we use to derive lower bounds using it as an example. We rely heavily on some basic results from the theory of approximations. These are detailed in Section A.2 of Appendix A. We then show how the technique generalises to other problems (Section 4.2). This involves a polynomial degree lower bound which is proved in Section 4.3. Section 4.4 deals with the optimality of the lower bounds derived, and Section 4.6 presents an optimal algorithm for a generalisation of the approximate medians problem. We then conclude with a discussion of our results and future directions.

4.1 Approximating the median

We begin by studying the problem of estimating the median of a sequence of numbers. (The problem described in Section 1.2 of the Introduction occurs as a special case of this problem.) This will help us illustrate the technique we use for deriving lower bounds for a more general class of problems.

Let $\epsilon > 0$ be any real. An *ϵ-approximate median* of a sequence $X = (x_0, \ldots, x_{n-1})$ of n numbers is a number x_i such that the number of x_j less than it and the number of x_j more than it are each less than $(1+\epsilon)\frac{n}{2}$. Roughly speaking, an ϵ-approximate median is any

number with rank in an interval of size ϵn around $\frac{n}{2}$. We may allow that the statistic not be a number from the input list. The problem of computing this quantity on a quantum computer was first studied by Grover with this relaxed definition [47, 48].

Suppose we are given a list X of n numbers as an oracle and we wish to compute an ϵ-approximate median by making as few queries to the oracle as possible. What is the least number of queries with which one can solve the problem? In 1996, Grover [47] gave an algorithm for this problem that makes $\tilde{O}(\frac{1}{\epsilon})$ queries when the numbers are all either 0 or 1. (When the numbers are drawn from a range of size M, the bound becomes $\tilde{O}(\frac{1}{\epsilon}\log M)$.) Here, the $\tilde{O}$ notation suppresses polylogarithmic factors. The restriction of the problem to 0/1 inputs is essentially the well-studied problem of estimating the bias of a random coin to within ϵ. It is known that it takes $\Theta(\frac{1}{\epsilon^2})$ coin tosses to solve this problem. Thus Grover's quantum algorithm provided further evidence of the speed-up possible with quantum computing. Could the algorithm be yet more efficient? A lower bound argument introduced by Bennett, Bernstein, Brassard and Vazirani [13] shows that the speed-up may be at most polynomial. More precisely, it shows that $\Omega(\frac{1}{\sqrt{\epsilon}})$ queries are necessary to compute an ϵ-approximate median (see also [73]). We now show that Grover's algorithm is essentially optimal, that, in fact, $\Omega(\frac{1}{\epsilon})$ queries are required.

Proving lower bounds for quantum algorithms is hard for exactly the same reasons that make the model powerful. The features of superposition and interference effects result in algorithms that explore exponentially many computational paths simultaneously and combine their results in non-trivial ways. Nevertheless, Beals, Buhrman, Cleve, Mosca and de Wolf [11] recently showed that the behaviour of quantum algorithms can effectively be captured in a simple, yet powerful manner by polynomials.

Lemma 4.1.1 (Beals,Buhrman,Cleve,Mosca,de Wolf) *Let $\mathcal{A}$ be a quantum algorithm that makes T calls to a boolean oracle X. Then, there is a real multilinear polynomial p of degree at most $2T$ in the variables $x_0, \ldots, x_{n-1}$ such that the acceptance probability of $\mathcal{A}$ on oracle input $X = (x_0, \ldots, x_{n-1}) \in \{0,1\}^n$ is exactly $p(x_0, \ldots, x_{n-1})$.*

Here, "acceptance probability" refers to the probability that the algorithm produces 1 as output. The power of this characterisation of quantum algorithms lies in the observation that if the algorithm computes some boolean function of the input X with probability at least $1-\delta$, then the corresponding polynomial *approximates* the function to within δ: if the function is 1 on some input, the algorithm returns 1 with probability at least $1-\delta$, and if

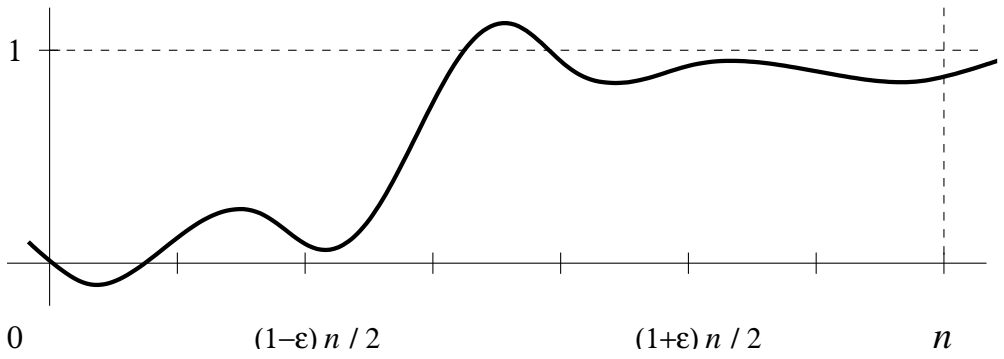

Figure 4.1: *The "projection" q of the polynomial p into one dimension.*

it is 0, the probability of acceptance (and hence the value of the polynomial at that point) may be at most δ. Thus, the task of proving a lower bound for the number of queries T reduces to that of proving a lower bound for the degree of polynomials approximating the function.

In the case of approximating the median, though, the restriction of the problem to $0,1$ inputs is not quite a function: the answer may be either 0 or 1 when the number of ones in the input list is in the range $((1-\epsilon)\frac{n}{2}, (1+\epsilon)\frac{n}{2})$. The polynomial p corresponding to any algorithm for the problem is therefore close to 0 if the number of ones is at most $(1-\epsilon)\frac{n}{2}$ and close to 1 if the number of ones is at least $(1+\epsilon)\frac{n}{2}$, but may assume any value in $[0,1]$ on other input points. It does not necessarily approximate any boolean function, and it would seem that we have reached a dead end. The observation that still allows us to use this characterisation is that where the output is uniquely defined, the polynomial depends essentially only on the number of ones in the input, and moreover, at points where the output may be either 0 or 1, the polynomial is undetermined, but nevertheless *bounded*. The polynomial still "approximates" a *symmetric boolean relation*. This observation may be exploited by using the technique of *symmetrisation*—averaging the value of the polynomial over all permutations of the input variables. The averaged polynomial clearly depends only on the number of ones in the input, is multilinear, and has degree at most that of the original polynomial. Fact A.2.1 from Section A.2 now allows us to reduce this polynomial to a *univariate* one. The result of this method of projecting the n-variate polynomial p to a univariate polynomial q is depicted in Figure 4.1.

We now are left with the problem of proving a degree lower bound for a univariate

polynomial q that has the following properties, assuming the algorithm makes error at most $\delta = \frac{1}{3}$ (this may be replaced by any constant less than a half):

- $q(i) \leq \frac{1}{3}$ for integers $0 \leq i \leq \lfloor (1-\epsilon)\frac{n}{2} \rfloor$,
- $q(i) \geq \frac{2}{3}$ for integers $\lceil (1+\epsilon)\frac{n}{2} \rceil \leq i \leq n$, and
- $0 \leq q(i) \leq n$ for all other integers in $[0, n]$.

Intuitively, it is clear that the polynomial q has "high" degree—it is close to zero initially, and makes a jump to one in a small interval around $\frac{n}{2}$, i.e., it has large derivative in that interval. Since a bounded low degree polynomial cannot have high derivative, we expect a large degree lower bound. A result from the theory of approximations, the Bernstein inequality (Fact A.2.5.2 in Section A.2), almost allows us to make this conclusion about q. The inequality says that if a polynomial is bounded (by, say, 1) in the interval $[-1, 1]$, then its derivative at any point x in the interval is bounded by $\frac{d}{\sqrt{1-x^2}}$, where d is the degree of the polynomial. The reason that this is not immediately applicable here is that polynomials such as arise from quantum algorithms may not be bounded at all points in the interval $[0, n]$, although they lie between 0 and 1 at integer points. In fact, they may have value exponential in the degree at non-integer points. To overcome this problem, we use a technique due to Paturi [63]. We consider a polynomial obtained by multiplying q by a low degree polynomial that "damps" the value of q outside a suitable interval. Since q may be exponentially large in its degree d, we use the approximately the dth power of the polynomial $(1 - x^2)$ suitably to achieve the damping. This polynomial mimics the function $\mathrm{e}^{-x^2 d}$ for large x in the interval $[-1, 1]$, and hence gives us the desired damping effect (see Figure 4.2).

We now formally state the above argument. We first scale and translate the polynomial q to transform the domain of interest from $[0, n]$ to $[-1, 1]$ to be able to apply the Bernstein inequality. Define $\hat{q}(x) = q((1+x)n/2)$. For $i = 0, 1, \ldots, n$, let $a_i = 2i/n - 1$. We may assume that $(1-\epsilon)\frac{n}{2}$ and $(1+\epsilon)\frac{n}{2}$ are integers, and that $\epsilon \leq \frac{1}{2}$. Define a polynomial r as

$$r(x) \;=\; \left[1 - \frac{(x-\epsilon)^2}{4}\right]^{cd},$$

for a constant c to be specified later. We will use this polynomial as the damping function. Let $s(x) = \hat{q}(x)r(x)$. The polynomial s lies between 0 and 1 at all a_i, $s(-\epsilon) \leq \frac{1}{3}$, $s(\epsilon) \geq \frac{2}{3}$,

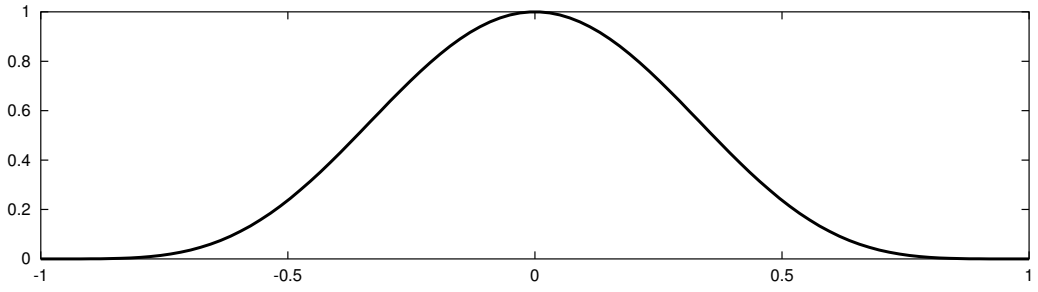

Figure 4.2: *The low degree polynomial* $(1-x^2)^t$ *mimics* $\mathrm{e}^{-x^2 t}$ *in the interval* $[-1,1]$.

and has degree $O(d)$. Furthermore, we will show that it is bounded by 1 suitably far from the origin. By Fact A.2.2, we have $|q(x)| \leq 2^d$ for all $x \in [0,n]$, hence the same bound applies to $\hat{q}$ and s in $[-1,1]$ (i.e., $\| s \| \leq 2^d$). Also, for $x \in [-1,-\frac{3}{4}] \cup [\frac{3}{4},1]$,

$$\begin{aligned} r(x) &= \left[1 - \frac{(x-\epsilon)^2}{4}\right]^{cd} \\ &\leq \exp\left[-\frac{(x-\epsilon)^2}{4} cd\right] \\ &\leq \exp(-cd/64), \end{aligned}$$

since $\epsilon \leq \frac{1}{2}$. By choosing $c = 64$, we ensure that s is bounded by 1 in the abovementioned interval.

It is now a straightforward matter to prove the degree bound. Suppose $\| s \| \leq 2$. The Mean Value Theorem implies that there is a point $a \in [-\epsilon, \epsilon]$ such that

$$\begin{aligned} s'(a) &\geq \frac{s(\epsilon) - s(-\epsilon)}{2\epsilon} \\ &\geq \frac{1}{6\epsilon}. \end{aligned}$$

On the other hand, by Bernstein inequality,

$$\begin{aligned} |s'(a)| &\leq \frac{(2c+1)d \| s \|}{\sqrt{1-a^2}} \\ &\leq \frac{258d}{\sqrt{3/4}}, \end{aligned}$$

which gives us an $\Omega(\frac{1}{\epsilon})$ lower bound for d. If $\| s \| > 2$, then $|s|$ attains its maximum in the interval $[-\frac{3}{4}, \frac{3}{4}]$. Since $|s|$ is bounded by 1 at points spaced $\frac{2}{n}$ apart, the Mean Value

Theorem implies that s has derivative at least

$$\frac{\| s \| - 1}{2/n} \geq \frac{n \| s \|}{4}$$

at some point $a \in [-\frac{3}{4}, \frac{3}{4}]$. Another application of Bernstein inequality gives us

$$\begin{aligned} |s'(a)| &\leq \frac{(2c+1)d \| s \|}{\sqrt{1-a^2}} \\ &\leq \frac{129d \| s \|}{\sqrt{7/16}}, \end{aligned}$$

which implies that $d = \Omega(n)$. We summarise our conclusion below.

Theorem 4.1.2 *The quantum query complexity of computing an ϵ-approximate median is $\Omega(\min\{\frac{1}{\epsilon}, n\})$.*

A natural measure of complexity of computing order statistics (such as the minimum and the median) is the number of *comparisons* between the input elements required for the computation. To study this aspect of such problems, one considers algorithms in the *comparison tree model*. In this model, the algorithm is provided with an oracle that replies with the result of the comparison $x_i < x_j$ when given a pair of indices (i, j), rather than an oracle that returns the number x_i on a query i as considered above. The query complexity of a problem is then the number of *comparisons* required to solve the problem. The lower bound for estimating the median continues to hold in the comparison tree model, since any comparison between two input numbers can be simulated by making at most four queries to an oracle of the sort we consider above.

4.2 Other problems of interest

The techniques we used to derive the degree lower bound in the previous section are quite general, and may be extended to derive lower bounds for several other problems of a statistical nature that have been studied recently. These include approximating the mean and approximate counting, for which only weaker lower bounds were known before. We define these problems now and then state the more general polynomial degree lower bound that enables us to infer their complexity.

Let $X = (x_0, \ldots, x_{n-1})$ be a sequence of (rational) numbers, and let $\epsilon, \Delta > 0$ be (rational) parameters. We may assume ϵ to be in the range $[\frac{1}{2n}, 1)$ and Δ to be in $[\frac{1}{2}, n)$. We define the following statistics:

1. **Δ-approximate kth-smallest element**. (Defined for $1 \leq k \leq n$.) A number x_i that is the jth-smallest element of X for some j in the range $(k - \Delta, k + \Delta)$.

2. **ϵ-approximate mean**. A number μ such that $|\mu - \mu_X| < \epsilon$, where $\mu_X = \frac{1}{n}\sum_{i=0}^{n-1} x_i$ is the mean of the n input numbers.

3. **Δ-approximate count**. (Defined when $x_i \in \{0, 1\}$ for all i.) A number t such that $|t - t_X| < \Delta$, where $t_X = |X| = \sum_{i=0}^{n-1} x_i$ is the number of ones in X.

4. **ϵ-approximate relative count**. (Defined when $x_i \in \{0, 1\}$ for all i.) A number t such that $|t - t_X| < \epsilon t_X$, where t_X is defined as above.

Some of these statistics are very closely related to each other. The first is a natural generalisation of an approximate median. The third is a scaled version of the second restricted to $0, 1$ inputs, and the last is a version of it in which we are interested in bounding *relative* rather than *additive* error. As in the case of approximate medians, we may relax the condition that the approximate statistic be a number from the input list (with a suitable modification to the definition above); our results continue to hold with the relaxed definition.

The feature that the above quantities have in common with approximate median is that when the sequence of input numbers is restricted to be over $\{0, 1\}$, they correspond to certain boolean relations. Thus, as in the case of the median problem, we may prove lower bounds for algorithms computing these statistics by showing degree bounds for polynomials approximating these relations. We make this precise below.

Consider a boolean relation $f : \{0, 1\}^n \to 2^{\{0,1\}}$. We say a real n-variate polynomial p *approximates* the relation f to within c, for a constant $0 \leq c < 1/2$, if

1. for all $X \in \{0, 1\}^n$, $p(X) \in [-c, 1 + c]$, and

2. for all points X at which $f(X) \in \{0, 1\}$, $|p(X) - f(X)| \leq c$.

Our main theorem, which we prove in Section 4.3, gives a degree lower bound for polynomials approximating relations of the of the following type. For $X = (x_0, \ldots, x_{n-1}) \in \{0, 1\}^n$, let $|X| = \sum_{i=0}^{n-1} x_i$ be the number of ones in X. Let ℓ, ℓ' be integers such that $0 \leq \ell \neq \ell' \leq n$. Define the boolean relation $f_{\ell,\ell'}$ on $\{0, 1\}^n$ as

$$f_{\ell,\ell'}(X) \quad = \quad \begin{cases} 1 & \text{if } |X| = \ell \\ 0 & \text{if } |X| = \ell' \\ \{0, 1\} & \text{otherwise} \end{cases}$$

Let $m \in \{\ell, \ell'\}$ be such that $\left|\frac{n}{2} - m\right|$ is maximised, and let $\Delta_\ell = |\ell - \ell'|$.

Theorem 4.2.1 *Let p be any real n-variate polynomial which approximates the boolean relation $f_{\ell,\ell'}$ to within c, for some constant $c < 1/2$. Then, the degree of p is $\Omega(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$.*

This theorem generalises a degree lower bound given by Paturi [63] for polynomials approximating symmetric *total* boolean functions.

We say that an algorithm $\mathcal{A}$ *computes* a relation f if $\Pr[\mathcal{A}(X) \notin f(X)] \leq \delta$ for all inputs X, where δ is some constant less than $1/2$. For boolean f, we say that the algorithm *accepts* an input X if $\mathcal{A}(X) = 1$. When combined with the characterisation due to Beals *et al.* (Lemma 4.1.1) of the acceptance probability of a quantum algorithm on a boolean input oracle, in terms of polynomials, Theorem 4.2.1 gives us the following result.

Corollary 4.2.2 *Any quantum black-box algorithm that computes the boolean relation $f_{\ell,\ell'}$, given the input as an oracle, makes $\Omega(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$ queries.*

Proof: Consider an oracle quantum algorithm $\mathcal{A}$ that computes the relation $f_{\ell,\ell'}$ with constant error probability $c < 1/2$ by making at most T oracle queries. From Lemma 4.1.1, there is a real multilinear polynomial $p(x_0, \ldots, x_{n-1})$ of degree at most $2T$ that gives the acceptance probability of $\mathcal{A}$ on the oracle input $X = (x_0, \ldots, x_{n-1})$. Clearly, p approximates $f_{\ell,\ell'}$ to within c: $p(X) \geq 1-c$ when $|X| = \ell$, $p(X) \leq c$ when $|X| = \ell'$, and $p(X) \in [0,1]$ for all $X \in \{0,1\}^n$. Theorem 4.2.1 now immediately implies the result. ■

Corollary 4.2.2 enables us to prove lower bounds for the query complexity of computing the statistics defined above by showing reductions from relations of the sort described above. The details of the reductions are given in Appendix A.3. We start with the generalisation of an approximate median defined above.

Theorem 4.2.3 *Any quantum black-box algorithm for computing a Δ-approximate kth-smallest element given n numbers via an oracle makes at least $\Omega(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$ queries.*

This lower bound also holds for the number of comparisons required (for the same reason as given for approximate medians).

Grover [48] recently gave an $O(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon})$ query algorithm for approximating the mean. This is an almost quadratic improvement over classical algorithms. When the inputs

are restricted to be $0, 1$, the problem reduces to that of approximate counting to within an additive error. We give a lower bound for this problem and in the process get an almost tight lower bound for the general mean estimation problem.

Theorem 4.2.4 *Any quantum black-box algorithm that approximates the number of ones of a boolean oracle to within an additive error of Δ makes $\Omega(\sqrt{n/\Delta}+\sqrt{t(n-t)}/\Delta)$ queries on inputs with t ones.*

Corollary 4.2.5 *$\Omega(\frac{1}{\epsilon})$ queries are required to compute an ϵ-approximate mean in the quantum black-box model.*

Brassard *et al.* [23, 60, 21] study the version of the approximate counting problem in which one is interested in bounding the *relative* error of the estimate. We show that their algorithm is optimal to within a constant factor (when the number of ones is $\leq (1-\epsilon)n$).

Theorem 4.2.6 *Any quantum black-box algorithm that solves the ϵ-approximate relative count problem makes*

$$\Omega\left(\sqrt{\frac{n}{\epsilon t+1}}+\frac{\sqrt{t(n-t)}}{\epsilon t+1}\right)$$

queries on input oracles with t ones.

4.3 A degree lower bound for polynomials

This section is devoted to deriving the polynomial degree lower bound stated in Theorem 4.2.1, which gives a lower bound for polynomials approximating symmetric relations. The bound is derived along the lines described in Section 4.1 using ideas employed by Paturi [63] for polynomials that approximate symmetric boolean functions.

Recall from Section 4.2 that $f_{\ell,\ell'}(X)$ is a boolean relation on $\{0,1\}^n$ which is 1 when $|X| = \ell$ and 0 when $|X| = \ell'$, that m is the one of ℓ and ℓ' which is furthest from $n/2$, and that $\Delta_\ell = |\ell - \ell'|$. We assume that p is an n-variate polynomial of degree d which approximates the relation f to within $1/3$ in the sense defined in Section 4.2. (The constant $1/3$ may be replaced by any constant less than $1/2$ and the proof continues to hold with minor changes.) Without loss of generality, we assume that $\ell > \ell'$. Otherwise, we work with the polynomial $1-p$, which approximates $1-f$.

We begin by replacing p with its *symmetrisation* p^{sym} and then using Fact A.2.1 to transform it to an equivalent *univariate* polynomial q. (Since $x^2 = x$ for $x \in \{0,1\}$,

we may assume that p is *multilinear*.) We show a degree lower bound for q, thus giving a degree lower bound for p.

In order to apply the derivative inequalities above, we transform the polynomial q to an equivalent polynomial $\hat{q}$ over the interval $[-1,1]$, where $\hat{q}(x) = q((1+x)n/2)$. For $i = 0, 1, \ldots, n$, let $a_i = 2i/n - 1$. Clearly, $\hat{q}$ has the following properties:

1. $\hat{q}$ has degree at most d.
2. $|\hat{q}(a_i)| \leq 4/3$ for $0 \leq i \leq n$.
3. $\hat{q}(a_\ell) \geq 2/3$ and $\hat{q}(a_{\ell'}) \leq 1/3$. Thus, by the Mean Value Theorem, there is a point a in the interval $[a_{\ell'}, a_\ell]$ such that $\hat{q}'(a) \geq (2/3 - 1/3)/(a_\ell - a_{\ell'}) = n/(6\Delta_\ell)$.

We prove two lower bounds for d, which together imply the theorem. The first of the lower bounds follows by applying the Markov Inequality (Fact A.2.5.1) directly to $\hat{q}$.

Lemma 4.3.1 $d = \Omega(\sqrt{n/\Delta_\ell}\,)$.

Proof: We consider two cases:
Case (a). $\| \hat{q} \| < 2$. Combining property 3 of $\hat{q}$ listed above and Fact A.2.5.1, we get

$$d^2 \;\geq\; \hat{q}'(a)/\| \hat{q} \| \;\geq\; n/(12\Delta_\ell).$$

So $d = \Omega(\sqrt{n/\Delta_\ell}\,)$.
Case (b). $\| \hat{q} \| \geq 2$. From property 2 of $\hat{q}$ listed above, every point at which $\hat{q}$ attains its norm is no more than $2/n$ away from a point a_i at which $|\hat{q}(x)| \leq 4/3$. Hence, by the Mean Value Theorem, there is a point $\hat{a} \in [-1,1]$ such that

$$|\hat{q}'(\hat{a})| \;\geq\; (\| \hat{q} \| - 4/3)/(2/n) \;\geq\; n \| \hat{q} \| /6.$$

The Markov inequality then implies $d = \Omega(\sqrt{n}\,) = \Omega(\sqrt{n/\Delta_\ell}\,)$. ■

The second of the lower bounds follows from an application of the Bernstein Inequalities for algebraic and *trigonometric* polynomials (Facts A.2.5.2 and A.2.6, respectively).

Lemma 4.3.2 $d = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

Proof: If $\hat{q}$ has norm less than 2, property 3 in conjunction with Fact A.2.5.2 implies that

$$2d \;\geq\; \| \hat{q} \| \, d \;\geq\; \sqrt{1-a^2}\, \hat{q}'(a) \;\geq\; \sqrt{1-a^2}\, (n/6\Delta_\ell).$$

But since $a \in [a_{\ell'}, a_\ell]$, we have

$$1 - a^2 \;\geq\; 1 - a_m^2 \;=\; 1 - (2m/n - 1)^2 \;=\; 4m(n-m)/n^2.$$

Hence, $d = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

Now suppose that $\| \hat{q} \| \geq 2$. The proof in this case is not as straightforward as in Lemma 4.3.1, since Fact A.2.5.2 gives a bound which is sensitive to the point at which $\hat{q}$ has high derivative. However, it is possible to "damp" the value of the polynomial outside a suitable interval, and thus obtain the required bound.

Let b be the point of smallest magnitude at which $|\hat{q}| \geq 2$, and let c be the one of b and a_ℓ of smaller magnitude. Assume that $c \geq 0$. (The proof in the other case is similar.) Let C be a constant such that $0 < C < 0.01$. We distinguish between two cases.

Case (a). $c \leq 1 - C$. Define the polynomial r to be:

$$r(x) \;=\; \hat{q}(x + c)(1 - x^2)^{d_1}$$

where $d_1 = \lceil 6/C^2 \rceil \, d$. The degree D of r is clearly $O(d)$, so it suffices to prove the claimed lower bound for D.

Suppose $\| r \| < 2$. Then, $c = a_\ell$, $r(0) \geq 2/3$, and $r(a_{\ell'} - a_\ell) \leq 1/3$. By the Mean Value Theorem, there is a point $\hat{a} \in [a_{\ell'} - a_\ell, 0]$ such that $|r'(\hat{a})| = \Omega(n/\Delta_\ell)$. We may assume, without loss of generality, that $\Delta_\ell \leq n/4$, so that $\hat{a} \in [-1/2, 0]$. (Otherwise, the lower bound follows trivially.) By Fact A.2.5.2, we conclude that $D = \Omega(n/\Delta_\ell) = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

We now focus on the case when $\| r \| \geq 2$. We show in Claim 4.3.3 below that $|r(x)|$ is bounded by 1 for $C \leq |x| \leq 1$. This implies that $\| r \|$ is attained within $[-C, C]$. Note that $|r|$ is bounded by $4/3$ at points $a_i - c$ separated by $2/n$ in $[-C, C]$. Hence, there is a point $\hat{a} \in [-C, C]$ at which $|r'(\hat{a})| \geq n \| r \| /6$. Applying Fact A.2.5.2 to r at the point $\hat{a}$, we get $D = \Omega(n) = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

It only remains to prove the following claim to complete the analysis of Case (a).

Claim 4.3.3 *For all* $x \in [-1, -C] \,\cup\, [C, 1]$, $|r(x)| \leq 1$.

Proof: Note that $\| \hat{q} \| = \max_{0 \leq x \leq n} |q(x)|$. By Fact A.2.2, we thus have $\| \hat{q} \| \leq (4/3) \cdot 2^d$. In particular, $|\hat{q}(x + c)| \leq (4/3) \cdot 2^d \leq (4/3) \cdot \mathrm{e}^{5d}$ for $x \in [-1, 1 - c]$. We give the same bound on $|\hat{q}(x + c)|$ for $x \in [1 - c, 1]$ by using Fact A.2.3:

$$|\hat{q}(x + c)| \;\leq\; \| \hat{q} \| \cdot T_d(x + c) \;\leq\; (4/3) \cdot 2^d \cdot \mathrm{e}^{2\sqrt{3}\, d} \;\leq\; (4/3) \cdot \mathrm{e}^{5d},$$

since $c \leq 1$. Further, if $C \leq |x| \leq 1$, we have $(1-x^2)^{d_1} \leq \mathrm{e}^{-x^2 d_1} \leq \mathrm{e}^{-6d}$. Combining these two inequalities, we may bound r in the region $[-1,-C] \cup [C,1]$ as follows:

$$|r(x)| \;=\; |\hat{q}(x+c)|\,(1-x^2)^{d_1} \;\leq\; (4/3)\cdot \mathrm{e}^{5d}\cdot \mathrm{e}^{-6d} \;\leq\; 1$$

■

We now turn to the remaining case, when c is close to 1.

Case (b). $c > 1 - C$. Without loss of generality, we assume that $\Delta_\ell \leq \ell', \ell \leq n - \Delta_\ell$. (Otherwise, the bound we seek follows from Lemma 4.3.1 above, since $\sqrt{m(n-m)}/\Delta_\ell \leq \sqrt{n/\Delta_\ell}$). This implies, in particular, that $c < 1$. Let $\alpha_c = \cos^{-1} c$. Since $0.99 < 1 - C < c < 1$, we have $0 < \alpha_c < 1/4$.

We prove a degree lower bound for a *trigonometric polynomial* s derived from $\hat{q}$. The polynomial s is defined as:

$$s(\theta) \;=\; \hat{q}(\cos\theta)[\cos(d_1(\theta-\alpha_c))]^{d_2},$$

where $d_1 = \lfloor 1/(2\alpha_c) \rfloor$ and $d_2 = c_1 \lceil d/d_1 \rceil$, for some integer constant $c_1 \geq 1$ to be specified later. Let D be the degree of the polynomial s.

Claim 4.3.4 $D = O(d)$.

Proof: First, note that since $\cos\theta \geq 1 - \theta^2/2$ for $\theta \in [0, \pi/2]$, we have

$$\alpha_c \;\geq\; \sqrt{1-\cos\alpha_c} \;=\; \sqrt{1-c} \;\geq\; \sqrt{2\Delta_\ell/n}.$$

(The last inequality follows from the assumption that $\ell \leq n - \Delta_\ell$.) Hence, $d_1 \leq 1/(2\alpha_c) = O(\sqrt{n/\Delta_\ell})$, which is $O(d)$ by Lemma 4.3.1. We may now bound D as follows:

$$D \;\leq\; d + d_2 d_1 \;=\; d + c_1 \lceil d/d_1 \rceil\, d_1 \;\leq\; d + c_1(d + d_1) \;=\; O(d).$$

■

Thus, it suffices to prove a lower bound for D of $\Omega(\sqrt{m(n-m)}/\Delta_\ell)$, which we do next.

Let $\alpha_i = \cos^{-1} a_i$, for $i = 0, \ldots, n$.

Again, if $\| s \| < 2$, we get the lower bound easily. In this case, $c = a_\ell$, $s(\alpha_\ell) \geq 2/3$, and $s(\alpha_{\ell'}) \leq 1/3$. Hence, for some $\alpha \in [\alpha_\ell, \alpha_{\ell'}]$, $|s'(\alpha)| \geq (1/3)/(\alpha_{\ell'} - \alpha_\ell)$. By the Mean Value Theorem, $\alpha_{\ell'} - \alpha_\ell = |\cos\alpha_{\ell'} - \cos\alpha_\ell| \,/\, \sin\hat{\alpha}$, for some $\hat{\alpha} \in [\alpha_\ell, \alpha_{\ell'}]$. Note that $\sin\hat{\alpha} \;\geq$

$\sin\alpha_\ell \geq \sin\alpha_m = \sqrt{1-a_m^2}$. Thus, $|s'(\alpha)| \geq (1/3)\sqrt{1-a_m^2}/(2\Delta_\ell/n)$. Fact A.2.6, the Bernstein Inequality for trigonometric polynomials, then gives us $D = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

We now examine the case when $\| s \| \geq 2$. Claim 4.3.5 below shows that $|s(\theta)|$ is bounded by 1 when $\theta \in [-\pi, -\pi+\alpha_c/2] \cup [-\alpha_c/2, \alpha_c/2] \cup [\pi-\alpha_c/2, \pi]$. We assume here that the norm is attained in $[0,\pi]$; the proof proceeds analogously in the other case. This point is close to some point $\alpha_i \in [\alpha_c/2, \pi-\alpha_c/2]$ where $|s(\alpha_i)| \leq 4/3$. Arguing as before, we get that for some points $\alpha, \beta \in [\alpha_c/2, \pi-\alpha_c/2]$, $|s'(\alpha)| \geq (\| s \| /3)(\sin\beta)/(2/n)$. Further,

$$\sin\beta \geq \sin\frac{\alpha_c}{2} \geq \frac{\alpha_c}{4} \geq \frac{\sin\alpha_c}{4} \geq \frac{\sin\alpha_m}{4}.$$

From Fact A.2.6, we now get $D = \Omega(\sqrt{m(n-m)}) = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

We now prove that s is bounded in the region mentioned above.

Claim 4.3.5 *For all $\theta \in [-\pi, -\pi+\alpha_c/2] \cup [-\alpha_c/2, \alpha_c/2] \cup [\pi-\alpha_c/2, \pi]$, $|s(\theta)| \leq 1$.*

Proof: We prove the claim for $\theta \in [0, \alpha_c/2]$. The analysis for θ in the other intervals is similar. (One exploits the fact that $\hat{q}(\cos\theta)$ is an *even* function of θ, and that Corollary A.2.4 limits its behaviour outside $[\alpha_c, \pi-\alpha_c]$.)

Let $h(\theta) = [\cos(d_1(\theta-\alpha_c))]^{d_2}$. Then, for $\theta \in [0, \alpha_c]$,

$$\begin{aligned} |h(\alpha_c-\theta)| = |\cos(d_1\theta)|^{d_2} &\leq (1-(d_1\theta)^2/4)^{d_2} \\ &\leq \mathrm{e}^{-d_2(d_1\theta)^2/4} \\ &\leq \mathrm{e}^{-c_1 d\theta^2/(16\alpha_c)}. \end{aligned}$$

The first inequality follows from the fact that $\cos\phi \leq 1-\phi^2/4$ for $\phi \in [0, \pi/2]$ and that $0 \leq d_1\alpha_c \leq 1/2$. In the last step, we use the fact that $\alpha_c \leq 1/4$.

Further, Corollary A.2.4 gives us the following bound on the value of $\hat{q}$ outside the interval $[-c, c]$:

$$|\hat{q}(c+x)| \leq 2\,|T_d(1+x/c)| \leq 2\cdot \mathrm{e}^{2d\sqrt{3x/c}},$$

for $x \in [0, 1-c]$. Since for $\theta \in [0, \alpha_c]$, $\cos(\alpha_c-\theta) = \cos\alpha_c\cos\theta + \sin\alpha_c\sin\theta \leq c + \alpha_c\theta$, we have

$$|\hat{q}(\cos(\alpha_c-\theta))| \leq 2\cdot \mathrm{e}^{2d\sqrt{3\alpha_c\theta/c}} \leq 2\cdot \mathrm{e}^{4d\sqrt{\alpha_c\theta}}.$$

Hence, for $\theta \in [0, \alpha_c/2]$,

$$|s(\theta)| = |\hat{q}(\cos(\alpha_c-(\alpha_c-\theta)))|\,|h(\alpha_c-(\alpha_c-\theta))| \leq 1,$$

provided c_1 is chosen large enough. ■

This completes the proof of Lemma 4.3.2. ■

Lemmas 4.3.1 and 4.3.2 together imply that

$$d = \Omega\left(\max\left\{\sqrt{n/\Delta_\ell}, \sqrt{m(n-m)}/\Delta_\ell\right\}\right),$$

which is equivalent to the bound stated in Theorem 4.2.1.

4.4 Optimality of the lower bounds

We saw in the previous sections that our lower bounds for the mean and the median come within logarithmic factors of known upper bounds, while in the case of approximate counting with relative error, the bound is optimal (for most of the inputs). An obvious question that arises is whether it is possible further optimize the algorithms or whether our lower bounds can be improved upon. We show new, optimal algorithms for some problems and also show that our methods for showing lower bounds cannot be improved.

In Section 4.6 we present an algorithm for approximating the kth-smallest element that comes within a constant factor of the optimum. This also gives us a new, optimal algorithm for estimating the median.

Theorem 4.4.1 *There is a quantum black-box algorithm that computes a Δ-approximate kth-smallest element of n numbers, using $O(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$ queries.*

Corollary 4.4.2 *$O(\frac{1}{\epsilon})$ queries are sufficient for computing an ϵ-approximate median in the black-box model.*

Our median algorithm is an improvement over the algorithm of Grover [47, 48] (which also depends on the size of the domain from which the input numbers are drawn). It achieves a quadratic speed up over classical algorithms in the worst case.

The the upper bounds given above for estimating the kth-smallest element and the median continue to hold in the comparison tree model. In particular, if we set $\Delta = 1$, we get an optimal $O(\sqrt{k(n-k+1)}\,)$ comparison algorithm for computing the kth-smallest element (c.f. Theorems 4.2.3 and 4.4.1). (An optimal $O(\sqrt{n})$ comparison algorithm was already known for computing the minimum of n numbers [39].) This should be contrasted with the bound of $\Theta(n)$ in the classical case [19].

Corollary 4.4.3 *Let $M = \sqrt{k(n-k+1)}$. Any comparison tree quantum algorithm that computes the kth-smallest element of a list of n numbers makes $\Omega(M)$ comparisons. Moreover, there is a quantum algorithm that solves this problem with $O(M)$ comparisons.*

A method due to Brassard *et al.* [23, 60, 21] for *exact* counting enables us to show that the computation of the mean can be made sensitive to the number of ones in the input, resulting in better bounds when $|t - n/2|$ is large. This algorithm is optimal to within a constant factor. (See Section 4.5 for more details.)

Theorem 4.4.4 *There is a quantum black-box algorithm that, given a boolean oracle input X, and an integer $\Delta > 0$, with high (constant) probability computes a Δ-approximate count and makes an expected $O(\sqrt{n/\Delta} + \sqrt{t(n-t)}/\Delta)$ number of queries on inputs with t ones.*

Using an approximate counting algorithm of Brassard *et al.* [23, 60, 21], we show in Section 4.5 that the query lower bound of Corollary 4.2.2 is optimal to within a constant factor.

Theorem 4.4.5 *The quantum query complexity of computing the relation $f_{\ell,\ell'}$, given the input as an oracle, is $O(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$.*

The result of Beals *et al.* (Lemma 4.1.1 above) then immediately implies that the degree lower bound of Theorem 4.2.1 is also optimal to within a constant factor.

Corollary 4.4.6 *For any constant $0 < c < 1/2$, there is a real, n-variate polynomial p of degree $O(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$ that approximates the relation $f_{\ell,\ell'}$ to within c.*

In view of this result, the remaining lower bounds derived in 4.2 cannot be improved using the methods we employ. In fact, we believe that the lower bounds for computing the mean and approximate counting with relative error are optimal, and that the upper bounds can be improved to match them (up to constant factors).

4.5 Algorithms based on counting

We first show how the relation $f_{\ell,\ell'}$ defined in Section 4.2 may be computed optimally, i.e., within a constant factor of the lower bound of Corollary 4.2.2, thus proving Theorem 4.4.5.

Our algorithm actually computes the relation $\hat{f}_{\ell,\ell'} : \{0,1\}^n \to \{0,1\}$, where $0 \leq \ell' < \ell \leq n$, defined as:

$$\hat{f}_{\ell,\ell'} = \begin{cases} 1 & \text{if } |X| \geq \ell \\ 0 & \text{if } |X| \leq \ell' \\ \{0,1\} & \text{otherwise} \end{cases}$$

Clearly, any algorithm for this relation also computes $f_{\ell,\ell'}$.

The algorithm $D(X, \ell', \ell)$ for $\hat{f}_{\ell,\ell'}$, which we call a *distinguisher*, is in fact an immediate derivative of an approximate counting algorithm of Brassard *et al.* [23, 60, 21], which enables us to estimate the number of ones t_Y of a boolean function Y in a useful manner.

Theorem 4.5.1 (Brassard, Høyer, Mosca, Tapp) *There is a quantum black-box algorithm $C(Y, P)$ which, given oracle access to a boolean function $Y = (y_0, \ldots, y_{n-1})$, and an explicit integer parameter P, makes P calls to the oracle Y and computes a number $t \in [0, n]$ such that*

$$|t_Y - t| \leq \frac{\sqrt{t_Y(n - t_Y)}}{P} + \frac{|n - 2t_Y|}{4P^2}$$

with probability at least $2/3$.

Let X be the input to the distinguisher D, and let m and Δ_ℓ be defined as in Section 4.2. Further, let $P = \left\lceil c(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell) \right\rceil$, where c is a constant to be specified later, and let $t = C(X, P)$. The algorithm $D(X, \ell', \ell)$ returns 0 if $t < \ell' + \Delta_\ell/2$ and 1 otherwise. The correctness of the algorithm follows from the claim below; its optimality is clear from the choice of P.

Claim 4.5.2 *With probability at least $2/3$, if $t_X \leq \ell'$, then $t < \ell' + \Delta_\ell/2$, and if $t_X \geq \ell$, then $t > \ell' + \Delta_\ell/2$.*

We give the proof of this claim in Section A.4 of the appendix. We will see in Section 4.6 that this distinguishing capability of D also helps us search for an element of a desired rank optimally.

Next, we sketch an optimal algorithm for approximate counting to within an additive error. Since both its form and analysis are identical to the exact counting algorithm of Brassard *et al.* [23, 60, 21], we omit the details.

Recall from Section 4.2 that the problem of computing a Δ-approximate count consists of computing a number in $[0, n]$ which is within an additive error of Δ from the number of ones t_X of a given boolean oracle input $X = (x_0, \ldots, x_{n-1})$.

The algorithm first invokes the procedure $C(X, P)$ of Theorem 4.5.1 a few times (say, five times), with $P = \left\lceil c\sqrt{n/\Delta} \right\rceil$ (for some suitable constant c), to get an estimate $\tilde{t}$, taken to be the median of the approximate counts returned by C. With high (constant) probability, this estimate is within $O(\min\{t_X, n - t_X\} + \Delta)$ of the actual count t_X. The algorithm then invokes C again, with $P = \left\lceil c_1(\sqrt{n/\Delta} + \sqrt{\tilde{t}(n-\tilde{t})}/\Delta) \right\rceil$ (for a suitable constant c_1) and outputs the value returned by C. It is easy to verify that with high (constant) probability, the approximate count obtained is within the required range. An analysis similar to that of the exact counting algorithm mentioned above yields the bound of Theorem 4.4.4 on the expected number of queries made by the algorithm.

4.6 Optimal approximate selection

Consider the problem of approximating the kth-smallest element in the black-box model. Recall that when provided with a list of numbers $X = (x_0, \ldots, x_{n-1})$ as an oracle, and an explicit parameter $\Delta > 1/2$, the task is to find an input number x_i (or the corresponding index i) such that x_i is a jth-smallest element for a $j \in (k-\Delta, k+\Delta)$. Notice that we may round Δ to $\lceil \Delta \rceil$ without changing the problem to be solved. We therefore assume that Δ is an integer in the sequel.

The description of the problem in terms of *ranks* of numbers needs to be given carefully, since there may be repeated numbers in the list. To accommodate repetitions, we let $\mathrm{rank}(x_i)$ denote the *set* of positions $j \in \{1, \ldots, n\}$ at which x_i could occur, when the list X is arranged in non-decreasing order. A Δ-approximate kth-smallest element is thus a number x_i such that $\mathrm{rank}(x_i) \cap (k - \Delta, k + \Delta)$ is non-empty. For ease of exposition, we will use "$\mathrm{rank}(x_i)$ is at least j" to mean "$\mathrm{rank}(x_i) \cap [1, j) = \emptyset$" and "$\mathrm{rank}(x_i)$ is at most j" to mean "$\mathrm{rank}(x_i) \cap (j, n] = \emptyset$."

In this section we give an optimal (up to a constant factor) quantum black-box algorithm for computing a Δ-approximate kth-smallest element. Our algorithm is inspired by the minimum finding algorithm of Dürr and Høyer [39]. It builds upon a generalisation of the Grover search algorithm [46] due to Boyer, Brassard, Høyer and Tapp [20] and the distinguisher described in Section 4.4 obtained from the approximate counting algorithm of Brassard, Høyer, Mosca and Tapp [23, 60, 21]. To compute an ϵ-approximate median within the bound stated in Corollary 4.4.2, one need only run the algorithm with the parameters k and Δ chosen appropriately.

4.6.1 An abstract algorithm

We first present the skeleton of our algorithm using two hypothetical procedures S and K. For convenience, we define $x_{-1} = -\infty$, and $x_n = \infty$. The procedure S takes a pair of inputs $-1 \leq i < j \leq n$ and returns an index chosen uniformly at random from the set of indices l such that $x_i < x_l < x_j$, if such an index exists. The procedure K takes an input $0 \leq i < n$ and returns 'yes' when x_i is a Δ-approximate kth-smallest element of X, '$<$' if x_i has rank at most $k - \Delta$ and '$>$' if x_i has rank at least $k + \Delta$. Our algorithm, which we refer to as $\mathcal{A}(S, K)$, performs a search on the list of input numbers, with a random pivot. It thus has the following form:

1. $i \leftarrow -1$, $j \leftarrow n$.
2. $\ell \leftarrow S(i, j)$.
3. If $K(\ell)$ returns 'yes', output x_ℓ (and/or ℓ) and stop.

   Else, if $K(\ell)$ returns '$<$', $i \leftarrow \ell$, go to step 2.

   Else, if $K(\ell)$ returns '$>$', $j \leftarrow \ell$, go to step 2.

Clearly, this algorithm always terminates and produces a correct solution. Since the running time (which we identify with the number of oracle queries made) of the subroutines with which we will replace S and K will depend on their inputs, we first analyse the probability that a given number in the input list is *ever* selected in step 2 of the algorithm.

Lemma 4.6.1 *Consider any fixed arrangement of the numbers $x_0, \ldots, x_{n-1}$ in sorted order and let x_{-1} be the 0th and x_n the $(n+1)$th element in this order. Let p_l be the probability that the input number in position l in this sorted list is ever selected in step 2 of the algorithm.*

1. *If $k \leq \Delta$, then for all $l \geq k + \Delta$, $p_l \leq \frac{1}{l+1}$,*
2. *if $k \geq n - \Delta$, then for all $l \leq k - \Delta$, $p_l \leq \frac{1}{n-l+1}$, and*
3. *if $\Delta < k < n - \Delta$, then*

$$p_l \leq \begin{cases} \frac{1}{k+\Delta-l} & \text{if } l \leq k - \Delta \\ \frac{1}{l+\Delta-k} & \text{if } l \geq k + \Delta \end{cases}$$

We defer the proof of this and all subsequent lemmas to Section A.4 of the appendix.

Our implementation of the procedure K will be somewhat different from the description given above. It will be closer to a randomised procedure $K'(\cdot)$ with the following specification. On input i (for some $0 \leq i < n$):

- if x_i is a $\frac{\Delta}{2}$-approximate kth-smallest element, output 'yes';
- else, if $\text{rank}(x_i)$ is at most $k - \Delta$, output '$<$';
- else, if $\text{rank}(x_i)$ is at least $k + \Delta$, output '$>$';
- else, if $\text{rank}(x_i)$ is at least $k - \Delta + 1$ and at most $k - \frac{\Delta}{2}$, probabilistically output either 'yes' or '$<$';
- else, if $\text{rank}(x_i)$ is at least $k + \frac{\Delta}{2}$ and at most $k + \Delta - 1$, probabilistically output either 'yes' or '$>$'.

The algorithm $\mathcal{A}(S, K')$ obtained by replacing the subroutine $K(\cdot)$ by $K'(\cdot)$ clearly also always computes a correct solution, although it may require more iterations of steps 2 and 3 to do so. However, we show that the probability of an input number being selected in step 2 of the algorithm does not increase by very much and thus that the expected running time of the algorithm remains of the same order.

Lemma 4.6.2 *Let X be any input oracle. The probability that an element of X is ever picked in step 2 of $\mathcal{A}(S, K')$ with oracle X and parameter Δ is at most the probability that it is ever picked in step 2 of $\mathcal{A}(S, K)$ on input X and $\Delta/2$.*

To prove the optimality of our algorithm, we will require bounds on the expected number of input elements (with rank in certain ranges) that are picked by the procedure S. We derive these next. Let q_i be the probability that x_i is ever picked in step 2 of the algorithm $\mathcal{A}(S, K')$. For x_i with $\text{rank}(x_i) \cap (k - \frac{\Delta}{2}, k + \frac{\Delta}{2}) \neq \emptyset$, we define $t_i = -1$. For any other x_i, let t_i be the $t \geq 0$ such that $\text{rank}(x_i) \cap (k - 2^t\Delta, k + 2^t\Delta) \neq \emptyset$ but $\text{rank}(x_i) \cap (k - 2^{t-1}\Delta, k + 2^{t-1}\Delta) = \emptyset$.

Lemma 4.6.3 *Let N_t be the expected number of elements x_i with $t_i = t$ picked by the sampling procedure S in step 2 of the algorithm $\mathcal{A}(S, K')$. Then, $N_{-1} \leq 1$ and $N_t \leq 4$ for $t \geq 0$.*

We are now ready to analyse the expected runtime (number of queries) made by our algorithm. Our implementation of S will be faithful to the specification above and will have an expected runtime of order of

$$\sqrt{\frac{n}{j-1-i}}$$

on input i, j. However, we will not be able to implement K' exactly. Our implementation K'' of K' will have the property that on input x_i with $t_i = t \geq 1$, with probability $1 - 2^{-t-5}$ it will output the correct answer with a runtime of order

$$\frac{t}{2^{t/2}}\sqrt{\frac{n}{\Delta}} + \frac{t}{2^t}\frac{\sqrt{k(n-k)}}{\Delta}.$$

For the rest of the x_i (with $t_i = 0, -1$), the performance is the same as that for $t_i = 1$.

We now bound the number of queries made by our algorithm.

Lemma 4.6.4 *The expected number of queries made by $\mathcal{A}(S, K'')$ conditioned on the event that K'' does not make any error in any invocation is of order*

$$\sqrt{\frac{n}{\Delta}} + \frac{\sqrt{k(n-k)}}{\Delta}.$$

Thus, the algorithm $\mathcal{A}(S, K'')$ terminates in optimal expected number of queries provided that K'' does not make any error.

Finally, we show that the probability that our algorithm makes an error is a small constant.

Lemma 4.6.5 *The probability that $\mathcal{A}(S, K'')$ either does not terminate or terminates with an incorrect result is at most $\frac{1}{4}$.*

Again, we postpone giving a proof till Appendix A.4.

By Markov's inequality, the previous two lemmas imply that if we run our algorithm for a suitably large constant times the expected number of queries, it terminates and gives the correct answer with probability at least 2/3. This completes the analysis of the algorithm, thus proving Theorem 4.4.1. Finally, we point out that our implementations of S and K' will access the input numbers only via comparisons, and thus may be adapted to work in the comparison tree model with the same bound on the number of oracle queries.

4.6.2 A realisation of the algorithm

We now show that it is possible to devise algorithms S and K'' as claimed in the previous section. The subroutine S is derived from the generalised search algorithm of Boyer *et al.* [20], which enables us to sample uniformly from the set of ones of a boolean function.

Theorem 4.6.6 (Boyer, Brassard, Høyer, Tapp) *There is a quantum black-box algorithm which, given a boolean oracle $Y = (y_0, \ldots, y_{n-1})$ with $|Y| \geq t \geq 1$, returns an index i chosen uniformly at random from the set $\{j \mid y_j = 1\}$ with $O(\sqrt{n/t}\,)$ expected queries.*

The procedure $S(i, j)$ is implemented by defining a boolean function $Y = (y_0, \ldots, y_{n-1})$, with $y_l = 1$ if and only if $x_i < x_l < x_j$, and using the sampling procedure above. Each "query" to the function Y requires four queries to the input oracle X. Thus, our sampling procedure satisfies the requirements of Section 4.6.1.

The procedure $K''(i)$ is realised by using the distinguisher D of Section 4.4 repeatedly to identify t_i (as defined in Section 4.6.1) by looking at both the number of elements smaller and the number of elements larger than x_i. As an intermediate step, we define a subroutine $R(d, i)$ which with probability $2/3$ meets the specifications of K' given in the previous section, but with the parameter Δ replaced by d.

Note that the probability of correctness of D may be boosted to $1 - 2^{\Omega(T)}$ by repeating the algorithm $O(T)$ times and returning the majority answer. We use this method to reduce its error probability to a small constant. We can now describe $R(d, i)$ as:

1. If $k + d - 1 > n$, go to step 2. Let $t_0 = \lceil k + d/2 \rceil - 2$, and $t_1 = k + d - 1$. Note that $0 \leq t_0 < t_1 \leq n$, since $k, d \geq 1$. Define a boolean function Y over a domain of size n, with $y_j = 1$ if and only if $x_j < x_i$. If the distinguisher $D(Y, t_0, t_1)$ returns '0', go to step 2. Otherwise, output '>'.

2. If $k - d < 0$, return 'yes'. Let $t_0 = n - \lfloor k - d/2 \rfloor - 1$, and $t_1 = n - k + d$. Note that we again have $0 \leq t_0 < t_1 \leq n$. Define a boolean function Y over a domain of size n, with $y_j = 1$ if and only if $x_j > x_i$. If the distinguisher $D(Y, t_0, t_1)$ returns '0', output 'yes'. Otherwise, output '<'.

It is easy to verify that this subroutine behaves as we require and makes $O(\sqrt{n/d} + \sqrt{k(n-k)}/d)$ queries to the oracle X. We are now ready to spell out the details of $K''(i)$:

1. Let t be the smallest number such that $[1, n] \subseteq (k - 2^t\Delta, k + 2^t\Delta)$.

2. While $t \geq 0$ do

   A. Repeat $R(2^t\Delta, i)$ independently $\Theta(t)$ times and consider the majority answer.

   B. If the answer is '$<$' or '$>$', stop and return that answer. Else, $t \leftarrow t - 1$.

3. $t = -1$. Return 'yes'.

Note that if the boosted version of R in step A does not make any error and $t_i \geq 0$, then it returns 'yes' until $t = t_i$ and then returns either 'yes' or '$<$' or '$>$' probabilistically. On $t = t_i - 1$, however, it correctly returns '$<$' or '$>$' for $t_i > 0$ and 'yes' for $t_i = 0$. For x_i with $t_i = -1$, the output is always 'yes' if there is no error. It is now a matter of some simple algebra to verify that this satisfies the properties of K'' stated in Section 4.6.1.

4.7 Concluding remarks

The polynomial method has served as a powerful paradigm in classical complexity theory [12]. As we saw in this chapter, the it has proved to be a powerful paradigm in quantum complexity theory as well (see also [11]). It has more recently also been applied to show a tight trade-off between query complexity and probability of error for the search problem [25]. No other method is known to give tight bounds for all ranges of error. A natural question is then whether this method *always* gives optimal bounds for quantum computation. The answer for *classical* computation is, of course, false (consider the search problem, for example).

While the polynomial method is very promising, proving degree lower bounds is technically very difficult. For the problems we studied, we were able to meaningfully transform a multivariate polynomial to a univariate one. Approximation by univariate polynomials is a well-studied subject, but no corresponding (useful) tools are known for reasoning about multivariate polynomials. It is as yet unclear whether the method can yield interesting bounds in more complex situations such as that of monotone functions, or symmetric non-boolean functions. Interestingly, it has been shown [4] that most n-variate boolean functions cannot be approximated by polynomials of degree less than $\approx \frac{n}{4}$, but degree bounds for explicit functions of interest elude us. It is also known that the degree of the polynomial exactly representing a boolean function is within the sixth power of that of

polynomials approximating it [11] (better bounds are known for monotone and symmetric functions). However, no better than a quadratic separation between these is known.

A challenging problem in the area of black-box computation is that of detecting collisions in cryptographic hash functions. These are functions $h : \{0,1\}^n \to \{0,1\}^n$ which are, say, two-to-one, and such that it is computationally infeasible to find two points x, y that collide (i.e., have $h(x) = h(y)$). Such "collision free" hash functions are of importance in designing digital signature schemes (see, e.g., [35]). Given such a hash function h as a black-box, any classical algorithm for finding collisions evaluates h at $\Theta(\sqrt{N})$ points, where $N = 2^n$. It is possible to improve the upper bound to $\sqrt[3]{N}$ using a quantum computer [22]. It is still unknown, though, whether these collision-free hash functions exist in the face of a quantum adversary—no non-trivial lower bound has been shown for the problem.

A breakthrough was recently made by Ambainis [6] who gave a new lower bound argument (a sophisticated version of a technique of Grover [49]) that gives a simple, unified explanation for most known lower bounds (including those reported in this chapter), while also improving several others. The argument may be called a "quantum adversary argument"—one considers a *superposition* of input oracles as an adversary and runs an algorithm on this input. The initially unentangled oracle state ends up entangled with the state of the algorithm at the end of the computation. By bounding the increase in entanglement due to each successive oracle query, a lower bound is obtained. It is possible that this approach will yield a meaningful lower bound for the collision problem as well.

Bibliography

[1] L. Adleman, J. Demarrais and M. Huang. Quantum computability. *SIAM Journal on Computing* **26**(5) (1997), pp. 1524–1540.

[2] D. Aharonov, A. Kitaev and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computation,* 1997, pp. 20–30.

[3] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, 1997, pp. 176–188.

[4] A. Ambainis. A note on quantum black-box complexity of almost all Boolean functions. *Information Processing Letters* **71** (1999), pp. 5–7.

[5] A. Ambainis. A better lower bound for quantum algorithms searching an ordered list. In *Proceedings of 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 352–357. IEEE.

[6] A. Ambainis. Quantum lower bounds by quantum arguments. Manuscript, 1999.

[7] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In *Proceedings of the 39th Symposium on Foundations of Computer Science,* 1998, pp. 332–341. IEEE.

[8] A. Ambainis, A. Nayak, A. Ta-Shma and U. Vazirani. Dense quantum coding and a lower bound for 1-way quantum automata. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, 1999, pp. 376–383.

[9] A. Ambainis, L.J. Schulman, A. Ta-Shma, U. Vazirani and A. Wigderson. The quantum communication complexity of sampling. In *Proceedings of the 39th Symposium on Foundations of Computer Science,* 1998, pp. 342–351. IEEE.

[10] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin and H. Weinfurter. Elementary gates for quantum computation. *Physical Review Letters A* **52** (1995), pp. 3457–3467.

[11] R. Beals, H. Buhrman, R. Cleve, M. Mosca and R. de Wolf. Quantum lower bounds by polynomials. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, 1998, pp. 352–361. IEEE.

[12] R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the 8th Annual IEEE Conference on Structure in Complexity Theory*, 1993, pp. 82–95.

[13] C. Bennett, E. Bernstein, G. Brassard and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing* **26**(5) (1997), pp. 1510–1523.

[14] C. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters* **70**, 1993, pp. 1895–1899.

[15] C. Bennett and S. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Physical Review Letters* **69** (1992), pp. 2881–2884.

[16] E. Bernstein and U.V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing* **26**(5) (1997), pp. 1411–1473.

[17] A. Berthiaume and G. Brassard. The quantum challenge to structural complexity. In *Proceedings of the 7th Annual IEEE Conference on Structure in Compexity*, 1992, pp. 132–137.

[18] A. Berthiaume and G. Brassard. Oracle quantum computing. In *Proceedings of the Workshop on Physics and Computation*, 1992, pp. 195–199.

[19] M. Blum, R.W. Floyd, V. Pratt, R.L. Rivest and R.E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences* **7** (1973), pp. 448–461.

[20] M. Boyer, G. Brassard, P. Høyer and A. Tapp. Tight bounds on quantum searching. *Forschritte Der Physik* **46** (1998), pp. 493–505.

[21] G. Brassard, P. Høyer, M. Mosca and A. Tapp. Quantum amplitude amplification and estimation. Manuscript, 1998.

[22] G. Brassard, P. Høyer and A. Tapp. Quantum algorithm for the collision problem. LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9705002`, May 1997.

[23] G. Brassard, P. Høyer and A. Tapp. Quantum counting. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pp. 820–831. Springer-Verlag, 1998.

[24] H. Buhrman, R. Cleve and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, 1998, pp. 63–68.

[25] H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 358–368. IEEE.

[26] H. Buhrman and R. de Wolf. Communication complexity lower bounds by polynomials. Computing Research Repository, `http://xxx.lanl.gov/abs/cs.CC/9910010`, October 1999.

[27] B. Chor, O. Goldreich, E. Kushelivitz and M. Sudan. Private information retrieval. *Proceedings of the 36th Symposium on Foundations of Computer Science,* 1995, pp. 41–50. IEEE. To appear in *Journal of the ACM.*

[28] I.L. Chuang. Personal communication, 1997.

[29] I.L. Chuang, L.M.K. Vandersypen, X. Zhou, D.W. Leung and S. Lloyd. Experimental realization of a quantum algorithm. *Nature* **393** (1998), pp. 143–146.

[30] J.I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Physics Review Letters* **74** (1995), pp. 4091–4094.

[31] R. Cleve, W. van Dam, M. Nielsen and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Proceedings of the 1st International Conference on Quantum Computing and Quantum Communication*, volume 1509 of *Lecture Notes in Computer Science*, pp. 61–74. Springer-Verlag, 1998.

[32] G.D. Cohen. A nonconstructive upper bound on covering radius. *IEEE Transactions on Information Theory* **IT-29**(3) (1983), pp. 352–353.

[33] D.G. Cory, A.F. Fahmy and T.F. Havel. Ensemble quantum computing by nuclear magnetic resonance spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America* **94** (1997), pp. 1634–1639.

[34] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, New York, 1991.

[35] I. Damgård. Collision free hash functions and public key signature schemes. In *Proceedings of EUROCRYPT'87*, volume 304 of *Lecture Notes in Computer Science*, pp. 203–216. Springer-Verlag, 1988.

[36] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London* **A400** (1985), pp. 96–117.

[37] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London* **A425** (1989), pp. 73–90.

[38] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London A* **439** (1992), pp. 553–558.

[39] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9607014`, July 1996.

[40] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser. A limit on the speed of quantum computation in determining parity. *Physics Review Letters* (81) (1998), pp. 5442–5444.

[41] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser. A limit on the speed of quantum computation for insertion into an ordered list. LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9812057`, December 1998.

[42] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser. How many functions can be distinguished with k quantum queries? LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9901012`, January 1999.

[43] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics* **21**(6-7) (1982), pp. 467–488.

[44] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences* **59**(2) (1999), pp. 240–252.

[45] N.A. Gershenfeld and I.L. Chuang. Bulk spin-resonance quantum computation. *Science* **275** (1997), pp. 350–356.

[46] L.K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, 1996, pp. 212–219.

[47] L.K. Grover. A fast quantum mechanical algorithm for estimating the median. LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9607024`, July 1996.

[48] L.K. Grover. A framework for fast quantum mechanical algorithms. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 1998, pp. 53–62.

[49] L.K. Grover. How fast can a quantum computer search? LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9809029`, September 1998.

[50] A.S. Holevo. Some estimates of the information transmitted by quantum communication channels. *Problemy Peredachi Informatsii* **9**, 1973, pp. 3–11. English translation in *Problems of Information Transmission* **9**, 1973, pp. 177–183.

[51] R.J. Hughes, G.L. Morgan and C.G. Peterson. Practical quantum key distribution over a 48-km optical fiber network. *Journal of Modern Optics*, to appear. Also available at the LANL Quantum Physics Archive, `http://xxx.lanl.gov/abs/quant-ph/9904038`, April 1999.

[52] Y. Ishai and E. Kushilevitz. Improved upper bounds on information-theoretic private information retrieval. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, 1999, pp. 79–88.

[53] J.A. Jones, M. Mosca and R.H. Hansen. Implementation of a quantum search algorithm on a nuclear magnetic resonance quantum computer. *Nature* **393** (1998), pp. 344–346.

[54] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proceedings of the 38th Symposium on Foundations of Computer Science,* 1997, pp. 66–75. IEEE.

[55] I. Kremer. *Quantum communication.* M.Sc. thesis, The Hebrew University of Jerusalem, 1995.

[56] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[57] K.-J. Lange, P. McKenzie and A. Tapp. Reversible space equals deterministic space. To appear in *Theoretical Computer Science*. Preliminary version in *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, 1997.

[58] M. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 2nd edition, 1988.

[59] C. Moore and J. Crutchfield. Quantum automata and quantum grammars. LANL Quantum Physics Archive, `http://xxx.lanl.gov/archive/quant-ph/9707031`, July 1997.

[60] M. Mosca. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *Proceedings of the Workshop on Randomized Algorithms, Mathematical Foundations of Computer Science*, 1998.

[61] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proceedings of the 40th Symposium on Foundations of Computer Science,* 1999, pp. 369–376. IEEE.

[62] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, 1999, pp. 384–393.

[63] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, 1992, pp. 468–474.

[64] A. Peres. *Quantum theory: concepts and methods.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[65] P.P. Petrushev and V.A. Popov. *Rational approximation of real functions.* Cambridge University Press, 1987.

[66] J. Preskill. Lecture notes for Physics 229: quantum information and computation. Available at `http://www.theory.caltech.edu/people/preskill/ph229`.

[67] M.O. Rabin. Probabilistic automata. *Information and Control* **6** (1963), pp. 230–245.

[68] R. Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing,* 1999, pp. 358–367.

[69] T.J. Rivlin. *The Chebyshev polynomials.* John Wiley and Sons, 1974.

[70] P.W. Shor. Fault-tolerant quantum computation. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 1996, pp. 56–65. IEEE.

[71] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**(5) (1997), pp. 1484–1509.

[72] D.R. Simon. On the power of quantum computation. *SIAM Journal on Computing* **26**(5) (1997), pp. 1474–1483.

[73] U.V. Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London, Series A* **356** (1998), pp. 1759–1768.

[74] J. Watrous. Space-bounded quantum complexity. *Journal of Computer and System Sciences.* To appear. A preliminary version appeared in *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, 1998, pp. 210–227.

[75] J. Watrous. PSPACE has constant-round quantum interactive proof systems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 112–119. IEEE.

[76] J. Watrous. On quantum and classical space-bounded processes with algebraic transition amplitudes. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 341–351. IEEE.

[77] A. Wehrl. General properties of entropy. *Reviews of Modern Physics* **50**(2) (1978), pp. 221–260.

[78] A.C.-C. Yao. Quantum circuit complexity. In *Proceedings of the 34th Symposium on Foundations of Computer Science*, 1993, pp. 352–361. IEEE.

Appendix A

Background, definitions and details of proofs

A.1 Information theory basics

We start with some concepts from classical information theory, state some of their useful properties and then turn to their quantum counterparts.

The *Shannon entropy* $H(X)$ of a classical random variable X that takes values x in some finite set with probability p_x is defined as

$$H(X) \;=\; -\sum_x p_x \log p_x.$$

The *mutual information* $I(X:Y)$ of a pair of random variables X, Y is defined by

$$I(X:Y) \;=\; H(X) + H(Y) - H(XY),$$

and the *conditional* entropy of X with respect to Y is

$$H(X|Y) \;=\; H(X) - I(X:Y).$$

We also use $H : [0,1] \to [0,1]$ to denote the binary entropy function

$$H(p) \;=\; -p \log p - (1-p)\log(1-p).$$

The following are basic properties of Shannon entropy. For any random variables X, Y, Z,

$$\begin{aligned} H(X|YZ) &\leq H(X|Y) \\ H(XY|Z) &\leq H(X|Z) + H(Y|Z). \end{aligned}$$

For other equivalent definitions and properties of these concepts, we refer the reader to a standard text (such as [34]) on information theory.

The quantum mechanical analogue of a random variable is a probability distribution over superpositions, also called a *mixed state*. Consider the mixed state $\{p_i, |\phi_i\rangle\}$, where the superposition $|\phi_i\rangle$ is drawn with probability p_i. The behaviour of this mixed state is completely characterised by its *density matrix* $\boldsymbol{\rho} = \sum_i p_i |\phi_i\rangle\langle\phi_i|$. We will therefore identify a mixed state with its density matrix.

The following properties of density matrices are immediate from the definition. For any density matrix $\boldsymbol{\rho}$,

1. $\boldsymbol{\rho}$ is Hermitian, i.e., $\boldsymbol{\rho} = \boldsymbol{\rho}^\dagger$.
2. $\boldsymbol{\rho}$ has unit trace, i.e., $\mathrm{Tr}(\boldsymbol{\rho}) = \sum_i \boldsymbol{\rho}(i,i) = 1$.
3. $\boldsymbol{\rho}$ is positive semi-definite, i.e., $\langle\psi|\,\boldsymbol{\rho}\,|\psi\rangle \geq 0$ for all $|\psi\rangle$.

Thus, every density matrix is *unitarily diagonalisable* and has non-negative real eigenvalues that sum up to 1. The *von Neumann entropy* $S(\boldsymbol{\rho})$ of a density matrix $\boldsymbol{\rho}$ is defined as $S(\boldsymbol{\rho}) = -\sum_i \lambda_i \log \lambda_i$, where $\{\lambda_i\}$ is the multiset of all the eigenvalues of $\boldsymbol{\rho}$. In other words, $S(\boldsymbol{\rho})$ is the Shannon entropy of the distribution induced by the eigenvalues of $\boldsymbol{\rho}$ on the corresponding eigenvectors.

The density matrix corresponding to a mixed state with superpositions drawn from a Hilbert space $\mathcal{H}$ is said to have *support* in $\mathcal{H}$. First, we note that von Neumann entropy is always non-negative. Furthermore,

Fact A.1.1 *If $\boldsymbol{\rho}$ is a density matrix with support in a Hilbert space of dimension d, then its entropy $S(\boldsymbol{\rho})$ is at most $\log d$.*

This is because the probability distribution induced by the eigenvalues of $\boldsymbol{\rho}$ has support of size at most d. The Shannon entropy of any such distribution is at most $\log d$.

When a unitary operator U is applied to a mixed state, the corresponding density matrix $\boldsymbol{\rho}$ is transformed to $U\boldsymbol{\rho}U^\dagger$. Since the eigenvalues of $U\boldsymbol{\rho}U^\dagger$ are the same as those of $\boldsymbol{\rho}$, we conclude that entropy is invariant under unitary operations.

Fact A.1.2 *For any density matrix $\boldsymbol{\rho}$ and unitary operator U, we have $S(U\boldsymbol{\rho}U^\dagger) = S(\boldsymbol{\rho})$.*

If a mixed state $\boldsymbol{\rho}$ is measured according to an orthogonal set of projections $\{P_j\}$, it is easily verified that the resulting density matrix is given by $\sum_j P_j \boldsymbol{\rho} P_j$. When we make an orthogonal measurement on a mixed state, the the entropy of the system can only increase.

Fact A.1.3 *Let $\boldsymbol{\rho}$ be the density matrix of a mixed state in a Hilbert space $\mathcal{H}$ and let the set of orthogonal projections $\{P_j\}$ define a measurement in $\mathcal{H}$. Further, let $\rho' = \sum_j P_j \boldsymbol{\rho} P_j$ be the density matrix resulting from a measurement of the mixed state with respect to this observable. Then $S(\boldsymbol{\rho}') \geq S(\boldsymbol{\rho})$.*

It is not hard to see that this is in fact a consequence of the property of density matrices that the entropy of any random variable obtained by making a measurement on a mixed state is at least as much as the entropy of its density matrix. A proof of this (latter) property may be found in [64, Chapter 9, pp. 262–263].

For a comprehensive introduction to the concept of entropy and its properties, see, for instance, [66, 77].

Now, consider a system consisting of some number of quantum bits in mixed state X. One is often interested in parts of this larger system consisting of disjoint subsets $X_1, X_2, \ldots, X_k$ of the qubits. The subsystems $X_1, X_2, \ldots$ are then identified with the mixed states induced by X on the corresponding parts and we write $X = X_1 X_2 \cdots$. We can define the "mutual information" $I(X : Y)$ of two mixed states (subsystems) X, Y in analogy with classical mutual information: $I(X : Y) = S(X) + S(Y) - S(XY)$. Note that not all properties of classical mutual information carry over to the quantum case. For example, it is not true in general that $I(X : Y) \leq S(X)$. Nonetheless, some of the intuition we have about mutual information still applies. For example, we have for any mixed states X, Y, Z,

$$I(X : YZ) = I(X : Y) + I(XY : Z) - I(Y : Z) \tag{A.1}$$

$$I(X : YZ) \geq I(X : Y). \tag{A.2}$$

Equation (A.1) follows immediately from the definition and captures the change in mutual information of two communicating parties when they exchange some number of quantum bits. Equation (A.2), The proof of Equation (A.2) is much more involved; This inequality is in fact equivalent to the *strong sub-additivity property* of von Neumann entropy.

A.2 Some properties of polynomials

In this section, we present some properties of polynomials and define some concepts that we will use for our results.

The *symmetrisation* p^{sym} of a multivariate polynomial $p(x_0, \ldots, x_{n-1})$ is defined to be

$$p^{\text{sym}}(x_0, \ldots, x_{n-1}) \;=\; \frac{\sum_{\pi \in S_n} p(x_{\pi(0)}, \ldots, x_{\pi(n-1)})}{n!} \quad ,$$

where S_n is the set of permutations on n symbols.

If p is a multilinear polynomial of degree d, then p^{sym} is also a multilinear polynomial of degree d. Clearly, p^{sym} is a *symmetric* function. The following fact attributed to Minsky and Papert [58] says that there is a succinct representation for p^{sym} as a *univariate* polynomial.

Fact A.2.1 *If $p : R^n \to R$ is a multilinear polynomial of degree d, then there exists a polynomial $q : R \to R$, of degree at most d, such that $q(x_0+x_1+\cdots+x_{n-1}) = p^{\text{sym}}(x_0, \ldots, x_{n-1})$ for $x_i \in \{0, 1\}$.*

In the remainder of this section, we will deal only with univariate polynomials over the reals.

The properties of polynomials that we use involve the concept of the *uniform* or *Chebyshev norm* of a polynomial (denoted by $\| p \|$), which is defined as: $\| p \| = \max_{-1 \le x \le 1} |p(x)|$. We will refer to the uniform norm of a polynomial as simply the *norm* of the polynomial.

The first property we require is a bound on the value of a polynomial in an interval, given a bound on its values at *integer* points in the interval.

Fact A.2.2 *Let p be a polynomial of degree $d \le n$ such that $|p(i)| \le c$ for integers $i = 0, \ldots, n$. Then $|p(x)| \le 2^d \cdot c$ for all x in the interval $[0, n]$.*

This fact follows easily from an examination of the *Lagrange interpolation* for the polynomial p; the details are omitted.

The next fact bounds the value of a polynomial *outside* the interval $[-1, 1]$, in terms of its norm (i.e., its maximum value *inside* the interval $[-1, 1]$). Let $T_d(x) = \frac{1}{2}[(x + \sqrt{x^2-1})^d + (x - \sqrt{x^2-1})^d]$. This polynomial is known as the *Chebyshev polynomial* of degree d. Note that $|T_d|$ is an *even* function of x, and that $|T_d(1+x)| \le \mathrm{e}^{2\sqrt{2x+x^2}}$, for $x \ge 0$.

Fact A.2.3 *Let p be a polynomial of degree at most d. Then, for $|x| > 1$,*

$$|p(x)| \leq \| p \| \cdot |T_d(x)| .$$

A proof of this fact may be found in Section 2.7 of [69]. We require an easy corollary of this fact.

Corollary A.2.4 *Let p be a polynomial of degree at most d, with $|p(x)| \leq c$ for $|x| \leq a$, for some $a > 0$. Then, for all $|x| \geq a$,*

$$|p(x)| \leq c\,|T_d(x/a)|$$

At the heart of our lower bound proof is the following set of inequalities, due to Bernstein and Markov, which relate the size of the derivative p' of a polynomial p to the degree of p. Proofs of these may be found in Section 3.4 of [65] and Section 2.7 of [69].

Fact A.2.5 *Let p be a polynomial of degree d. Then, for $x \in [-1, 1]$,*

1. **(Markov)** $|p'(x)| \leq d^2 \| p \|$;
2. **(Bernstein)** $\sqrt{1 - x^2}\ |p'(x)| \leq d \| p \|$.

The next fact, which is a more general version of the Bernstein Inequality above, deals with *trigonometric polynomials*. A *trigonometric polynomial* $t(x)$ of degree d is a real linear combination of the functions $\cos ix$ and $\sin ix$, where i is an integer in the range $[0, d]$. For a trigonometric polynomial t, we define its norm to be $\| t \| = \max_{-\pi \leq x \leq \pi} |t(x)|$.

Fact A.2.6 *Let t be a trigonometric polynomial of degree d. Then, for $x \in [-\pi, \pi]$,*

$$|t'(x)| \leq d \| t \| .$$

A.3 Proofs of some black-box lower bounds

In this section, we show how to reduce from relation computations of the type given in Corollary 4.2.2 to approximating the kth-smallest element and to approximate counting, and we show how bounds for approximating the mean follow. In this way, we are able to show new quantum query lower bounds for the computation of these approximate statistics.

The following two lemmas specialise Corollary 4.2.2 to cases of interest to us. The first deals with relation $f_{\ell,\ell'}$ such that neither ℓ' nor ℓ is "close" to 0 or n, and the second covers the remaining case.

Lemma A.3.1 *Let $k, \Delta > 0$ be integers such that $2\Delta < k < n - 2\Delta$. Then, the quantum query complexity of $f_{k-\Delta,k+\Delta}$ is $\Omega(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$.*

Proof: We assume that $k \leq n/2$; the other case is symmetric. In applying Corollary 4.2.2, $\Delta_\ell = 2\Delta$. Since $k \leq n/2$, $m = k - \Delta$. Moreover, $(k-\Delta)(n-k+\Delta) > (k/2)(n-k)$. Corollary 4.2.2 now gives us the claimed bound. ■

Lemma A.3.2 *Let k, Δ be integers such that $0 < \Delta \leq n/4$ and $0 \leq k \leq 2\Delta$. Then, the quantum query complexity of $f_{0,k+\Delta}$ is $\Omega(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$. The same bound holds for $f_{k-\Delta,n}$ when $k \geq n - 2\Delta$.*

Proof: We prove the first part of the lemma; the other part is symmetric. In applying Corollary 4.2.2, $\Delta_\ell = k + \Delta \leq 3\Delta$ and $m = 0$. Hence, we get a bound of $\Omega(\sqrt{n/\Delta})$ for $f_{0,k+\Delta}$. For the lemma to hold, we need only show that the second term in the claimed lower bound is of the order of the first term: $\sqrt{k(n-k)}/\Delta \leq \sqrt{(2\Delta)n}/\Delta = O(\sqrt{n/\Delta})$. ■

We now prove the rest of the lower bound theorems of Section 4.2 by exhibiting reductions from suitable problems. We first consider the problem of estimating the kth-smallest element.

Proof of Theorem 4.2.3: We need only prove the bound when $\Delta \leq n/4$, since it holds trivially otherwise. We assume that Δ is integral. The same proof works with $\lceil \Delta \rceil$ substituted for Δ.

Note that the query complexity of computing $f_{\ell,\ell'}$ is the same as that of computing $f_{n-\ell,n-\ell'}$, since we can negate the oracle responses in an algorithm for the former to get an algorithm for the latter, and vice-versa. We now consider two cases:

Case (a). $2\Delta < k < n - 2\Delta$. Any algorithm for computing a Δ-approximate kth-smallest element also computes $f_{n-k+\Delta,n-k-\Delta}$, and hence, by Lemma A.3.1 and the observation above, it makes at least $\Omega(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$ queries.

Case (b). $k \leq 2\Delta$ or $k \geq n - 2\Delta$. If $k \leq 2\Delta$, we reduce from the relation $f_{n,n-k-\Delta}$. Lemma A.3.2, along with the observation above, gives the required bound. Similarly, for $k \geq n - 2\Delta$, we reduce from $f_{n-k+\Delta,0}$ to get the required bound. ■

The remaining proofs for approximate counting and approximating the mean are similar to the ones above; we only sketch them here.

Proof of Theorem 4.2.4: We may assume that $\Delta < n/8$, since the lower bound is trivial otherwise. Consider any algorithm that approximately counts to within an additive error of Δ. Fix any $0 \leq t \leq n$. Suppose for any input X with $|X| = t$, the algorithm outputs a Δ-approximate count after T queries with probability at least $2/3$. We then consider the truncated version of the algorithm which stops after making T queries and outputs 1 if the approximate count obtained (if any) lies in the range $(t - \Delta, t + \Delta)$ and 0 otherwise. Since the original algorithm approximates to within Δ for all inputs, the truncated algorithm computes $f_{t,t+\lceil 2\Delta \rceil}$ and/or $f_{t,t-\lceil 2\Delta \rceil}$ whenever these relations are well-defined (i.e., when $t + 2\Delta \leq n$ and/or $t - 2\Delta \geq 0$). Now, by considering the four cases $t \leq 4\Delta$, $n - t \leq 4\Delta$, $4\Delta < t \leq n/2$, and $n/2 < t < n - 4\Delta$, and by reducing from a suitable relation (either $f_{t,t+\lceil 2\Delta \rceil}$ or $f_{t,t-\lceil 2\Delta \rceil}$) in each case, we get the claimed lower bound. ■

Since the problem of approximate counting is a restriction of the more general problem of estimating the mean of n numbers, the lower bound for the latter problem follows directly from Theorem 4.2.4.

Proof of Corollary 4.2.5: If the input numbers are all 0/1, multiplying an ϵ-approximate mean by n gives us an ϵn-approximate count. From Theorem 4.2.4, in the worst case (when the number of ones in the input is $\lfloor n/2 \rfloor$), the number of queries required to solve the approximate mean problem is $\Omega(1/\epsilon)$. ■

Finally, we sketch the proof of the lower bound for approximate counting to within some relative error.

Proof of Theorem 4.2.6: To derive a lower bound on the number of queries T made by an algorithm to approximate t_X, when $t_X = t$, we consider a truncated version of the algorithm obtained by running the algorithm until it returns a value between $(1-\epsilon)t$ and $(1+\epsilon)t$ with probability at least $2/3$, for such inputs. Since the algorithm correctly approximates the count to within a relative error of ϵ for *all* inputs, we can use it to compute the relation $f_{t,t+1}$ when $\epsilon t \leq 1/4$, or $f_{t',t}$, where $t' = \lfloor (1-\epsilon)t/(1+\epsilon) \rfloor$, when $1/4 < \epsilon t$. Corollary 4.2.2 now gives us the claimed bound. ■

A.4 Proofs in the analysis of algorithms in Chapter 4

Proof of Claim 4.5.2: Recall that $m \in \{\ell, \ell'\}$ is such that $\left|\frac{n}{2} - m\right|$ is maximised, and

that $\ell' < \ell$. We prove the claim when $m \leq n/2$; the analysis of the other case is symmetric. Since $m \leq n/2$, $m = \ell'$. Theorem 4.5.1 says that with probability at least 2/3,

$$|t_X - t| \;\leq\; \frac{\sqrt{t_X(n-t_X)}}{P} + \frac{|n-2t_X|}{4P^2}.$$

Then, if $t_X \leq \ell' = m \leq n/2$, and if c is large enough,

$$\begin{aligned} |t - t_X| &< \frac{\sqrt{\ell' n}}{c\sqrt{\ell' n/2}/\Delta_\ell} + \frac{n}{4(c^2 n/\Delta_\ell)} \\ &< \frac{\Delta_\ell}{2}. \end{aligned}$$

In this case, $t \;<\; t_X + \Delta_\ell/2 \;\leq\; \ell' + \Delta_\ell/2$. At the same time, we also have $t \geq g(t_X)$, where $g(x)$ is the function

$$g(x) \;=\; x - \frac{\sqrt{xn}}{P} - \frac{n}{4P^2}.$$

We will show that g is an increasing function of x for $x \geq \ell$ and that $g(\ell) > \ell - \Delta_\ell/2 = \ell' + \Delta_\ell/2$, provided c is large enough. From this, it will follow that when $t_X \geq \ell$, $t \geq g(t_X) > \ell' + \Delta_\ell/2$, completing the proof.

For $x \geq \ell$, we have

$$\begin{aligned} g'(x) &= 1 - \frac{\sqrt{n}}{2P\sqrt{x}} \\ &\geq 1 - \frac{\sqrt{n}}{2P\sqrt{l}} \\ &\geq 1 - \frac{\sqrt{n}}{2c\sqrt{n/\Delta_\ell}\sqrt{\ell}} \;>\; 0, \end{aligned}$$

if c is large enough, since $\ell \geq \Delta_\ell$. Hence, g is increasing for all $x \geq \ell$. Moreover, if c is large enough, we have

1. $\frac{n}{4P^2} \;\leq\; \frac{n}{4(c^2 n/\Delta_\ell)} \;<\; \frac{\Delta_\ell}{4}$;
2. if $\ell' > \Delta_\ell$, then $\ell < 2\ell'$, $\frac{\sqrt{\ell n}}{P} \;\leq\; \frac{\sqrt{2\ell' n}}{(c\sqrt{\ell' n/2/\Delta_\ell})} \;<\; \frac{\Delta_\ell}{4}$;
3. if $\ell' \leq \Delta_\ell$, then $\ell \leq 2\Delta_\ell$, $\frac{\sqrt{\ell n}}{P} \;\leq\; \frac{\sqrt{2\Delta_\ell n}}{(c\sqrt{n/\Delta_\ell})} \;<\; \frac{\Delta_\ell}{4}$.

It follows from these observations, that

$$\begin{aligned} g(\ell) &= \ell - \frac{\sqrt{\ell n}}{P} - \frac{n}{4P^2} \\ &> \ell - \frac{\Delta_\ell}{2}. \end{aligned}$$

■

Proof of Lemma 4.6.1: We concentrate on case 3, when $\Delta < k < n - \Delta$. The analysis in the other two cases is similar.

For any r, s such that $0 \leq r < k - \Delta$ and $k + \Delta < s \leq n + 1$, let $p(l, r, s)$ denote the probability that the lth number in the sorted list is *ever* chosen in step 2 of the algorithm *after* i becomes the index of the rth number and j becomes the index of the sth number in the sorted list. We would like to bound $p_l = p(l, 0, n + 1)$ for all l in the range $[1, k - \Delta] \cup [k + \Delta, n]$. (The sum of the probability values p_l for $l \in (k - \Delta, k + \Delta)$ is clearly 1.)

Suppose $r + 1 < l \leq k - \Delta$. We get the following recurrence by considering the result of one invocation of S:

$$p(l, r, s) \quad \leq \quad \frac{1}{s - 1 - r} \left[\sum_{r'=r+1}^{l-1} p(l, r', s) + 1 + \sum_{s'=k+\Delta}^{s-1} p(l, r, s') \right].$$

The inequality is due to the possibility of repetitions in the input list. Furthermore, $p(l, l - 1, k + \Delta) \leq 1/(k + \Delta - l)$. By induction, we now get

$$p(l, r, s) \ \leq \ \frac{1}{k + \Delta - l},$$

for all $0 \leq r < l \leq k - \Delta$ and $k + \Delta \leq s \leq n + 1$. By a similar argument, we get

$$p(l, r, s) \ \leq \ \frac{1}{l + \Delta - k},$$

when $k + \Delta \leq l < s \leq n + 1$. ■

Proof of Lemma 4.6.2: Denote by a *run*, a possible sequence of indices generated by the procedure S in an execution of the algorithm $\mathcal{A}$. We compare runs of the algorithm $\mathcal{A}(S, K')$ with parameter Δ with runs of the algorithm $\mathcal{A}(S, K)$ with parameter $\Delta/2$. Observe that each run of $\mathcal{A}(S, K')$ or any prefix of it is also a prefix of runs of $\mathcal{A}(S, K)$, and that its probability is at most the sum of the probabilities of the runs of $\mathcal{A}(S, K)$ of which it is a prefix. The probability of an index being picked in step 2 of $\mathcal{A}(S, K')$ is the net probability of all the runs containing it. It is now not hard to see that this is bounded by the net probability of the runs of $\mathcal{A}(S, K)$ containing the index. ■

Proof of Lemma 4.6.3: Recall that q_i is the probability that x_i is ever picked in step 2 of the algorithm $\mathcal{A}(S, K')$. The expectation N_t is then exactly $\sum_{i:t_i=t} q_i$. Note that N_{-1} is at most 1 since the algorithm halts with certainty when it picks x_i with $t_i = -1$. We claim that for $t \geq 0$, this expectation is at most 4.

Let n_t be the number of input elements x_i with $t_i = t \geq 0$ with rank at most $k - 2^{t-1}\Delta$. For these elements, $q_i \leq \frac{1}{2^{t-1}\Delta}$. This holds because by Lemma 4.6.2, the probability is at most $\frac{1}{k+\Delta/2-l}$, assuming $k < n-\Delta$, and at most $\frac{1}{n-l+1}$ for other k for every $l \in \text{rank}(x_i)$. Choosing the largest such l gives us the claimed bound. Out of the elements singled out above, at most $2^{t-1}\Delta$ elements have rank greater than $k - 2^t\Delta$, so their contribution to N_t is at most 1. The remaining elements are all equal, and hence include $k - 2^{t-1}\Delta - n_t + 1$ in their rank. These numbers have $q_i \leq \frac{1}{n_t}$. (Since $q_i \leq p_l$ for all $l \in \text{rank}(x_i)$, we can use Lemma 4.6.2 with $l = k - 2^{t-1}\Delta - n_t + 1$ to get this bound.) Their contribution to N_t is thus also at most 1. We can similarly bound the contribution of q_i for x_i with $t_i = t$ but with rank at least $k + 2^{t-1}\Delta$ by 2. ■

Proof of Lemma 4.6.4: We calculate the expected number of queries made as follows. The queries made by subroutine K'' given index i as input are "charged" to i. The queries made by S on input i, j are charged to one of i, j such that the corresponding element is closest to k in rank.

The key observation is that given that no error occurs, K'' behaves exactly like K'. Let t_i and q_i be as defined in Section 4.6.1. Let $q_0 = q_{n+1} = 1$. Note that whenever x_i with $t_i = t \geq 0$ is charged something due to S, the number of queries charged is at most $O(\sqrt{n/(2^t\Delta)})$.

We may now bound the expected number of queries made by (omitting constant factors throughout)

$$\begin{aligned}
&\sum_{i=0}^{n+1} q_i \mathrm{E}\left[\text{queries charged to } x_i\right] \\
&\quad\leq \sum_{t\geq 1}\sum_{i:t_i=t} q_i \left[\sqrt{\frac{n}{2^t\Delta}} + \frac{t}{2^{t/2}}\sqrt{\frac{n}{\Delta}} + \frac{t}{2^t}\frac{\sqrt{k(n-k)}}{\Delta}\right] + \sqrt{\frac{n}{\Delta}} + \frac{\sqrt{k(n-k)}}{\Delta} \\
&\quad\leq \sum_{t\geq 1} N_t \left[\sqrt{\frac{n}{2^t\Delta}} + \frac{t}{2^{t/2}}\sqrt{\frac{n}{\Delta}} + \frac{t}{2^t}\frac{\sqrt{k(n-k)}}{\Delta}\right] + \sqrt{\frac{n}{\Delta}} + \frac{\sqrt{k(n-k)}}{\Delta} \\
&\quad\leq \sqrt{\frac{n}{\Delta}} + \frac{\sqrt{k(n-k)}}{\Delta}
\end{aligned}$$

which is precisely the bound claimed. ■

Proof of Lemma 4.6.5: The algorithm $\mathcal{A}(S, K'')$ makes an error only if K'' makes an error on some input i chosen in step 2 of the algorithm. Let E_i be the even that i is chosen

in step 2. Then,

$$\begin{aligned}
\Pr\left[\mathcal{A}(S, K'') \text{ errs}\right] &\leq \sum_i \Pr\left[E_i \text{ occurs and } K'' \text{ errs on } i \mid \text{ no error occurred before}\right] \\
&= \sum_i \Pr\left[K'' \text{ errs on } i\right] \cdot q_i \\
&= \sum_t \sum_{i:t_i=t} q_i \Pr\left[K'' \text{ errs on } i\right] \\
&\leq \frac{2}{64} + \sum_{t\geq 1} N_t \cdot 2^{-t-5} \\
&\leq \frac{1}{4}
\end{aligned}$$

which is the claimed error probability. ■