The quantum query complexity of approximating the median and related statistics *

Ashwin Nayak † *Felix Wu* ‡

Abstract

Let $X = (x_0, \ldots, x_{n-1})$ be a sequence of n numbers. For $\epsilon > 0$, we say that x_i is an ϵ-approximate median if the number of elements strictly less than x_i and the number of elements strictly greater than x_i are each less than $(1+\epsilon)\frac{n}{2}$. We consider the quantum query complexity of computing an ϵ-approximate median, given the sequence X as an oracle. We prove a lower bound of $\Omega(\min\{\frac{1}{\epsilon}, n\})$ queries for any quantum algorithm that computes an ϵ-approximate median with any constant probability greater than $1/2$. We also show how an ϵ-approximate median may be computed with $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log\log(\frac{1}{\epsilon}))$ oracle queries, which represents an improvement over an earlier algorithm due to Grover [11, 12]. Thus, the lower bound we obtain is essentially optimal. The upper and the lower bound both hold in the comparison tree model as well.

Our lower bound result is an application of the polynomial paradigm recently introduced to quantum complexity theory by Beals *et al.* [1]. The main ingredient in the proof is a polynomial degree lower bound for real multilinear polynomials that "approximate" symmetric *partial* boolean functions. The degree bound extends a result of Paturi [15] and also immediately yields lower bounds for the problems of approximating the kth-smallest element, approximating the mean of a sequence of numbers, and approximately counting the number of ones of a boolean function. All bounds obtained come within a polylogarithmic factor of the optimal (as we show by presenting algorithms where no such optimal or near optimal algorithms were known), thus demonstrating the power of the polynomial method.

*Part of this work was done when the first author was at the 1998 Elsag-Bailey – I.S.I. Foundation research meeting on quantum computation.

†Computer Science Division, UC Berkeley, Berkeley, CA 94720. Supported by JSEP grant FDF 49620-97-1-0220-03-98 and NSF grant CCR 9800024. Email: ashwin@cs.berkeley.edu.

‡Computer Science Division, UC Berkeley, Berkeley, CA 94720. Supported by an NDSEG Fellowship and NSF grant CCR 9800024. Email: felix@cs.berkeley.edu.

STOC '99 Atlanta GA USA

1 Introduction

1.1 Synopsis

Proving non-trivial lower bounds for any universal model of computation is a formidable task, and quantum computers are no exception to this. It is thus natural to seek bounds in restricted settings. The first such step in the field of quantum computation was taken by Bennett *et al.* [2]. They prove that we cannot solve NP-complete problems in sub-exponential time on a quantum computer merely by adopting the brute-force strategy of "guessing" solutions and checking them for correctness. Nonetheless, Grover's search algorithm [10] shows that a *quadratic* speed-up over classical algorithms is possible in this case. Thus, while the parallelism and the potential for interference inherent in quantum computation are not sufficient to significantly speed up certain strategies for solving problems, they *do* give some advantage over probabilistic computation. These results motivate the question as to whether similar speed up is possible in other scenarios as well.

Strategies such as "brute-force search" may formally be modelled via "black-box" computation, in which information about the input is supplied to the algorithm by an *oracle*. For example, the black-box search problem may be defined as follows: given oracle access to n bits $X = (x_0, \ldots, x_{n-1})$, compute an index i such that $x_i = 1$, if such an index exists. A simpler formulation would require a yes/no answer according to whether such an index exists or not. This amounts to computing the logical OR of the input bits. In the black-box setting, strategies are evaluated by studying the *query complexity* of the problem, i.e., the minimum number of oracle accesses needed in the worst case to solve the problem. In the case of the abstract search problem, the query complexity is the number of bits that need to be examined (in the worst case) in order to compute the logical OR of the n bits.

Considerable success has been achieved in the study of the query complexity of computing *boolean functions* in the quantum black box model, both in terms of optimal lower bounds for specific functions [2, 4, 9, 1], and in terms of general techniques for proving such lower bounds [2, 7, 1]. However, few approaches were known for the study of more general functions. Consider, for example, the problem of approximating the median of n numbers. 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 $(1+\epsilon)\frac{n}{2}$. The problem then is to compute such an x_i, given an oracle to the sequence X of input values, and an explicitly specified parameter $\epsilon > 0$ (which may be assumed to be at least $\frac{1}{2n}$). Grover gives an algorithm for finding an ϵ-approximate median that makes $\tilde{O}(\frac{1}{\epsilon})$ queries to the input oracle [11, 12]. (Here, the $\tilde{O}$ notation suppresses factors involving $\log(\frac{1}{\epsilon})$ and M, where M is the size of the domain the numbers are picked from.) Thus, an almost quadratic speed up over the best classical algorithm can be achieved (assuming M to be constant). However, it was still open whether this algorithm could be improved upon. In particular, known techniques such as the "hybrid argument" yield a lower bound of $\Omega(\frac{1}{\sqrt{\epsilon}})$ for the number of queries [18], whereas $O(\frac{1}{\epsilon})$ was suspected to be optimal. In this paper, we prove a lower bound of $\Omega(\frac{1}{\epsilon})$ for the query complexity of the approximate median problem, thus showing that Grover's algorithm is almost optimal. We also present a new $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log\log(\frac{1}{\epsilon}))$ query algorithm for the problem, thereby eliminating the dependence of the upper bound on M. The upper and the lower bound both also hold in the *comparison tree model*, in which one is interested in the number of *comparisons between the input elements* required to compute an ϵ-approximate median.

Our lower bound is derived via the *polynomial method* recently introduced to the area of quantum computing by Beals *et al.* [1]. 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 2/3, the corresponding polynomial *approximates* the function to within 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 *partial* boolean function, i.e., a function that is not necessarily defined at all points of its domain. Our result is thus based on a degree lower bound for polynomials that "approximate" symmetric partial boolean functions. This degree lower bound generalizes a bound due to Paturi [15], and also gives lower bounds for the problems of approximating the kth smallest element, approximating the mean of a sequence of numbers, and approximately counting the number of ones of a boolean function. All bounds obtained are tight or almost tight (as we show by presenting algorithms where no such optimal or near optimal algorithms were known), demonstrating the power of the polynomial method.

1.2 Summary of results

Consider a partial boolean function $f : \{0,1\}^n \to \{0,1\}$. We say a real n-variate polynomial p *approximates* the partial function 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 is defined, $|p(X) - f(X)| \leq c$.

Our main theorem gives a degree lower bound for polynomials approximating partial boolean functions 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 partial boolean function $f_{\ell,\ell'}$ on $\{0,1\}^n$ as

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

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

Theorem 1.1 *Let p be any real n-variate polynomial which approximates the partial boolean function $f_{\ell,\ell'}$ 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 generalizes a degree lower bound given by Paturi [15] for polynomials approximating symmetric *total* boolean functions.

We say that an algorithm $\mathcal{A}$ *computes* a *partial* function f on $\{0,1\}^n$, if $\mathbf{Pr}[\mathcal{A}(X) \neq f(X)] \leq \delta$ for all inputs X for which f is defined, 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 a characterization due to Beals *et al.* [1, Lemma 4.2] of the acceptance probability of a quantum algorithm on a boolean input oracle, in terms of polynomials, Theorem 1.1 gives us the following result.

Corollary 1.2 *Any quantum black-box algorithm that computes the partial boolean function $f_{\ell,\ell'}$, given the input as an oracle, makes $\Omega(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$ queries.*

This lower bound also holds for the *expected* query complexity of computing the partial function $f_{\ell,\ell'}$. Using an approximate counting algorithm of Brassard *et al.* [5, 14, 6], we show that our query lower bound is optimal to within a constant factor.

Theorem 1.3 *The quantum query complexity of computing the partial function $f_{\ell,\ell'}$, given the input as an oracle, is $O(\sqrt{n/\Delta_\ell} + \sqrt{m(n-m)}/\Delta_\ell)$.*

The result of Beals *et al.* mentioned above then immediately implies that the degree lower bound of Theorem 1.1 is also optimal to within a constant factor.

Corollary 1.4 *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 function $f_{\ell,\ell'}$ to within c.*

Corollary 1.2 enables us to prove lower bounds for the query complexity of computing the statistics listed below, given an oracle to a list $X = (x_0, \ldots, x_{n-1})$ of (rational) numbers in the range $[0, 1]$ and an explicitly specified real parameter $\epsilon > 0$ or $\Delta > 0$. We may assume ϵ to be in the range $[\frac{1}{2n}, 1)$ and Δ to be in $[\frac{1}{2}, n)$.

1. **ϵ-approximate median.** A number x_i such that $|\{j : x_j < x_i\}| < (1+\epsilon)\frac{n}{2}$ and $|\{j : x_j > x_i\}| < (1+\epsilon)\frac{n}{2}$.
2. **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 $(k - \Delta, k + \Delta)$.
3. **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.
4. **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.
5. **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 the problems defined above are very closely related to each other. Problem 2 is a natural generalization of problem 1; problem 4 is the restriction of problem 3 to boolean inputs (with Δ defined appropriately), and problem 5 is a version of problem 4 in which we are interested in bounding *relative* rather than *additive* error. In the case of problems 1 and 2, we may relax the condition that the approximate statistic be a number from the input list (with a suitable modification to definition 2 above); our results continue to hold with the relaxed definitions. (Problem 1 was first studied by Grover [11, 12] with this relaxed definition.)

We first prove a lower bound for approximating the kth-smallest element by showing reductions from partial functions of the sort described above. We thus get a lower bound for the approximate median problem as well.

Theorem 1.5 *Any quantum black-box algorithm for computing a Δ-approximate kth-smallest element makes at least $\Omega(\sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta)$ oracle queries.*

Corollary 1.6 *The quantum query complexity of computing an ϵ-approximate median is $\Omega(1/\epsilon)$.*

We also propose an algorithm for approximating the kth-smallest element that comes within a polylogarithmic factor of the optimum. This gives us a new algorithm for estimating the median. (We believe that it is possible to optimize the algorithm, but we do not attempt this here.)

Theorem 1.7 *Let $N = \sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta$. There is a quantum black-box algorithm that computes a Δ-approximate kth-smallest element of n numbers, using $O(N \log N \log\log N)$ queries.*

Corollary 1.8 *$O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon})\log\log(\frac{1}{\epsilon}))$ queries are sufficient for computing an ϵ-approximate median in the black-box model.*

Our median algorithm represents an improvement over the algorithm of Grover [11, 12] when the input numbers are allowed to be drawn from an arbitrarily large domain. The algorithm achieves an almost quadratic speed up over classical algorithms in the worst case.

A very natural measure of complexity of computing functions such as the kth-smallest element 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 returns 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. The query complexity of a problem is then the number of *comparisons* required to solve the problem. The lower and 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 almost optimal $\tilde{O}(\sqrt{k(n-k+1)})$ comparison algorithm for computing the kth-smallest element (c.f. Theorems 1.5 and 1.7). (An optimal $O(\sqrt{n})$ comparison algorithm was already known for computing the minimum of n numbers [8].) This should be contrasted with the bound of $\Theta(n)$ in the classical case [3].

Corollary 1.9 *Let $N = \sqrt{k(n-k+1)}$. Any comparison tree quantum algorithm that computes the kth-smallest element of a list of n numbers makes $\Omega(N)$ comparisons. Moreover, there is a quantum algorithm that solves this problem with $O(N \log N \log\log N)$ comparisons.*

Another application of Corollary 1.2 is to the problem of approximating the mean. Grover [12] recently gave an $O(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon})$ query algorithm for this problem, which is again an almost quadratic improvement over classical algorithms. When the inputs are restricted to be 0/1, the problem reduces to the counting problem. Using the approximate counting algorithm of Brassard *et al.* [5, 14, 6] mentioned above, we 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.

Theorem 1.10 *There is a quantum black-box algorithm that, given a boolean oracle input X, and an integer $\Delta > 0$, 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.*

We show that this algorithm is optimal to within a constant factor, and, in the process, we get an almost tight lower bound for the general mean estimation problem.

Theorem 1.11 *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 1.12 *The quantum query complexity of the ϵ-approximate mean problem is $\Omega(\frac{1}{\epsilon})$.*

Brassard *et al.* [5, 14, 6] 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 $t \leq (1-\epsilon)n$).

Theorem 1.13 *Any quantum black-box algorithm that solves the ϵ-approximate relative count problem makes*

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

queries on inputs with t ones.

In view of Corollary 1.4, the lower bounds stated above cannot be improved using the methods we employ in this paper. In fact, we believe that the lower bounds are optimal, and that the upper bounds can be improved to match them (up to constant factors).

2 The lower bound theorem and its applications

This section is devoted to deriving a polynomial degree lower bound, and to showing how lower bounds for the query complexity of the different black-box problems defined in Section 1.2 follow from it. We first prove the degree lower bound for polynomials in Section 2.1, and then apply the result to quantum black-box computation in Section 2.2.

2.1 A degree lower bound for polynomials

We now prove our main result, Theorem 1.1, which gives a lower bound for polynomials approximating symmetric partial functions. The bound is derived using a technique employed by Paturi [15] for polynomials that approximate non-constant symmetric boolean functions. Our bound generalizes and subsumes the Paturi bound.

We refer the reader to Appendix A for the definition of the concepts involved in the proof. Appendix A also summarizes the various facts about polynomials that we use to derive the bound.

Our proof rests heavily on the inequalities of Bernstein and Markov (Facts A.6 and A.5). The essence of these inequalities is that if a polynomial has a "large" derivative at a point suitably close to the origin, the polynomial has "high" degree.

Proof of Theorem 1.1: Recall from Section 1.2 that $f_{\ell,\ell'}(X)$ is a partial boolean function 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 partial function f to within $1/3$ in the sense defined in Section 1.2. (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 *symmetrization* p^{sym} and then using Fact A.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.5.1) directly to $\hat{q}$.

Lemma 2.1 $d = \Omega(\sqrt{n/\Delta_\ell})$.

Proof: We consider two cases:

Case (a). $\|\hat{q}\| < 2$. Combining property 3 of $\hat{q}$ listed above and Fact A.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

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

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.5.2 and A.6, respectively).

Lemma 2.2 $d = \Omega(\sqrt{m(n-m)}/\Delta_\ell)$.

Proof: If $\hat{q}$ has norm less than 2, property 3 in conjunction with Fact A.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 2.1, since Fact A.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.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 2.3 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.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 2.3 *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, we thus have $\|\hat{q}\| \leq (4/3) \cdot 2^d$. In particular, $|\hat{q}(x+c)| \leq (4/3) \cdot 2^d \leq (4/3) \cdot e^{5d}$ 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.3:

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

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

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

■

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 2.1 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 2.4 $D = O(d)$.

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 2.1. 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.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 2.5 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.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 2.5 *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.4 limits its behaviour outside $[\alpha_c, \pi - \alpha_c]$.)

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

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

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

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

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

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

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 2.2. ■

Lemmas 2.1 and 2.2 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 1.1. ■

2.2 Applications to quantum black-box computation

In this section, we use our degree lower bound in conjunction with a result of Beals *et al.* [1] to derive lower bounds for the quantum black-box complexity of approximating the statistics defined in Section 1.2. The key lemma of [1] which we require is the following:

Lemma 2.6 (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 $p(x_0, \ldots, x_{n-1})$ of degree at most $2T$ such that the acceptance probability of $\mathcal{A}$ on oracle input $X = (x_0, \ldots, x_{n-1})$ is exactly $p(x_0, \ldots, x_{n-1})$.*

We deduce Corollary 1.2 from Theorem 1.1 using this lemma.

Proof of Corollary 1.2: Consider an oracle quantum algorithm $\mathcal{A}$ that computes the partial function $f_{\ell,\ell'}$ with constant error probability $c < 1/2$ by making at most T oracle queries. From the lemma above, there is a 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 1.1 now immediately implies the result. ■

In the remainder of this section, we show how to reduce from partial function computations of the type given in Corollary 1.2 to approximating the kth-smallest element and to approximate counting, and we show how bounds for approximating the median and 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 specialize Corollary 1.2 to cases of interest to us. The first deals with functions $f_{\ell,\ell'}$ such that neither ℓ' nor ℓ is "close" to 0 or n, and the second covers the remaining case.

Lemma 2.7 *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 1.2, $\Delta_\ell = 2\Delta$. Since $k \leq n/2$, $m = k - \Delta$. Moreover, $(k - \Delta)(n - k + \Delta) > (k/2)(n - k)$. Corollary 1.2 now gives us the claimed bound. ■

Lemma 2.8 *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 1.2, $\Delta_\ell = k + \Delta \leq 3\Delta$ 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 1.2 by exhibiting reductions from suitable problems. We first consider the problem of estimating the kth-smallest element.

Proof of Theorem 1.5: 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 2.7 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 function $f_{n,n-k-\Delta}$. Lemma 2.8, 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. ■

Since the problem of approximating the median is really a special case of the more general problem of approximating the kth-smallest element, we get a lower bound for the approximate median problem as well.

Proof of Corollary 1.6: For n odd, an ϵ-approximate median is a Δ-approximate kth smallest element for $k = (n+1)/2$ and $\Delta = \lceil(\epsilon n+1)/2\rceil$. The lower bound of $\Omega(1/\epsilon)$ now follows from Theorem 1.5. ■

The lower bounds for estimating the median and the kth-smallest element continue to hold in the comparison tree model, since any comparison between two input numbers can be simulated by making at most 4 queries to an oracle of the sort we consider above.

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

Proof of Theorem 1.11: 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 $0 \le t \le n$. 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 partial functions are well-defined (i.e., when $t+2\Delta \le n$ and/or $t-2\Delta \ge 0$). Now, by considering the four cases $t \le 4\Delta$, $n-t \le 4\Delta$, $4\Delta < t \le n/2$, and $n/2 < t < n-4\Delta$, and by reducing from a suitable partial function (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 1.11.

Proof of Corollary 1.12: If the input numbers are all 0/1, multiplying an ϵ-approximate mean by n gives us an ϵn-approximate count. From Theorem 1.11, 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 1.13: 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 function $f_{t,t+1}$ when $\epsilon t \le 1/4$, or $f_{t',t}$, where $t' = \lfloor(1-\epsilon)t/(1+\epsilon)\rfloor$, when $1/4 < \epsilon t$. Corollary 1.2 now gives us the claimed bound. ■

3 Some optimal or nearly optimal algorithms

We now show that the quantum black-box bounds obtained in the previous section are either tight or almost tight by giving algorithms where no such algorithm was known.

3.1 An optimal distinguisher

Recall the problem of computing the partial function $f_{\ell,\ell'}$ defined in Section 1.2. In this section, we show how this partial function may be computed optimally, i.e., within a constant factor of the lower bound of Corollary 1.2, thus proving Theorem 1.3. Along with Lemma 2.6, this implies that the polynomial degree lower bound we show in Theorem 1.1 is within a constant factor of the optimal, and hence it is not possible to obtain better lower bounds for the problems we consider using our technique.

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

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

Clearly, any algorithm for this partial function also computes $f_{\ell,\ell'}$, and thus the lower bound for the latter also holds for this function.

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.* [5, 14, 6], which enables us to estimate the number of ones t_Y of a boolean function Y in a useful manner.

Theorem 3.1 (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*

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

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 1.2. 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 determined 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 3.2 *With probability at least $2/3$, if $t_X \le \ell'$, then $t < \ell' + \Delta_\ell/2$, and if $t_X \ge \ell$, then $t > \ell' + \Delta_\ell/2$.*

We omit the proof of this claim. We will see in the next section that this distinguishing capability of D also allows us to search for an element of a desired rank nearly optimally.

3.2 Approximating the kth-smallest element

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 function to be computed. 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 $\text{rank}(x_i)$ denote the *set* of positions $j \in \{0, \ldots, n-1\}$ 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 $\text{rank}(x_i) \cap (k-\Delta, k+\Delta)$ is non-empty.

In this section we give a near optimal quantum black-box algorithm for computing a Δ-approximate kth-smallest element. No non-trivial algorithm was known for this problem for general k. Our algorithm is inspired by the minimum finding algorithm of Dürr and Høyer [8], and builds upon the general search algorithm of Boyer *et al.* [4] and the distinguisher of the last section obtained from the approximate counting algorithm of Brassard *et al.* [5, 14, 6]. To compute an ϵ-approximate median within the bound stated in Corollary 1.8, one need only run the algorithm with the parameters k and Δ chosen appropriately.

3.2.1 An abstract algorithm

We first present the skeleton of our algorithm using two hypothetical procedures $S(\cdot,\cdot)$ and $K(\cdot)$. For convenience, we define $x_{-1} = -\infty$, and $x_n = \infty$. The procedure $S(i,j)$ 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 $K(i)$ returns 'yes' when x_i is a Δ-approximate kth-smallest element of X, '<' if x_i has rank at most $k-\Delta$ (i.e., $\text{rank}(x) \cap (k-\Delta, n] = \emptyset$) and '>' if x_i has rank at least $k+\Delta$ (i.e., $\text{rank}(x) \cap [1, k+\Delta) = \emptyset$). 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. $l \leftarrow S(i,j)$.
3. If $K(l)$ returns 'yes', output x_l (and/or l) and stop.
   Else, if $K(l)$ returns '<', $i \leftarrow l$, go to step 2.
   Else, if $K(l)$ returns '>', $j \leftarrow l$, go to step 2.

An execution of steps 2 and 3 is called a *stage*. This algorithm always terminates and produces a correct solution within $n - 2\Delta + 2$ stages. However, the following lemma tells us that the *expected* number of stages before termination is small. Let $N = \sqrt{n/\Delta} + \sqrt{k(n-k)}/\Delta$.

Lemma 3.3 *The algorithm $\mathcal{A}(S,K)$ terminates with success after an expected $O(\log N)$ number of stages.*

The proof of this lemma, which we omit, proceeds by examining, for each input number, the probability that it is *ever* selected in step 2 of the algorithm. The expected number of stages is the sum of these probabilities. Note that the lemma guarantees that, with probability at least 3/4, the algorithm $\mathcal{A}(S,K)$ terminates within $O(\log N)$ stages.

We now consider the behaviour of the algorithm $\mathcal{A}$ when the (deterministic) procedure $K(\cdot)$ is replaced by a randomized subroutine $K'(\cdot)$ with the following specification. On input i (for some $0 \le i < n$):

- 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 stages to arrive at a solution, we show that the increase is by at most a constant factor.

Lemma 3.4 *Let X be any input oracle. The expected number of stages of the algorithm $\mathcal{A}(S,K')$ with oracle X and parameter Δ is at most the expected number of stages of $\mathcal{A}(S,K)$ on inputs X and $\Delta/2$.*

We omit the proof of this lemma. In light of Lemma 3.3, this lemma implies that $\mathcal{A}(S,K')$ also terminates after an expected $O(\log N)$ number of stages.

Finally, we would like to allow the procedures S and K' to either report failure or output an incorrect answer with some small probability. As mentioned above, we can restrict the number of stages of the algorithm to $O(\log N)$ and yet achieve success with probability at least 3/4. Now, if any invocation of S or K' fails (or errs) with probability $o(1/\log N)$, the net probability of success will still be at least, say, 2/3.

3.2.2 A realization of the algorithm

We are now ready to give implementations of the two procedures S and K' out of which the algorithm is built.

The subroutine S is derived from the generalized search algorithm of Boyer *et al.* [4], which enables us to sample uniformly from the set of ones of a boolean function.

Theorem 3.5 (Boyer, Brassard, Høyer, Tapp) *There is a quantum black-box algorithm which, given a boolean oracle $Y = (y_0, \ldots, y_{n-1})$ with $|Y| \ge t$, makes $O(\sqrt{n/t})$ queries and returns an index i chosen uniformly at random from the set $\{j : y_j = 1\}$, with probability at least 2/3.*

Note that the success probability of the procedure above may be amplified to $1 - 2^{\Omega(T)}$ by repeating it at most $O(T)$ times and returning a sample as soon as a 'one' of Y is obtained. It can easily be verified that a sample so generated is

uniform over the ones of Y. 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. Every time S is invoked in $\mathcal{A}$, there are at least Δ ones in Y. Hence, we choose the parameter t in Theorem 3.5 to be Δ and the number of repetitions T to be $\Theta(\log\log N)$. Each "query" to the function Y requires two queries to the input oracle X. Thus, our sampling procedure makes $O(\sqrt{n/\Delta}\log\log N)$ queries and succeeds with probability $1 - o(1/\log N)$.

The subroutine $K'(i)$ is implemented by using the distinguisher D of Section 3.1 to look at both the number of elements smaller and the number of elements larger than x_i. 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 require that the probability of error be $o(1/\log N)$, so we take T to be $\Theta(\log\log N)$. In more detail:

1. If $k + \Delta - 1 > n$, go to step 2. Let $t_0 = \lceil k + \Delta/2 \rceil - 2$, and $t_1 = k + \Delta - 1$. Note that $0 \le t_0 < t_1 \le n$, since $k, \Delta \ge 1$. 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 $k - \Delta < 0$, return 'yes'. Let $t_0 = n - \lfloor k - \Delta/2 \rfloor - 1$, and $t_1 = n - k + \Delta$. Note that we again have $0 \le t_0 < t_1 \le n$. 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 meets the specification for K' with probability $1 - o(1/\log N)$, and that it makes $O(N\log\log N)$ queries to the oracle X.

By Lemma 3.4, we conclude that the total number of queries made by the algorithm is $O(N\log N\log\log N)$, as claimed in Theorem 1.7. Observe that our implementations of S and K' use only comparisons between the input numbers, and thus may be adapted to work in the comparison tree model, with the same bound on the number of oracle queries.

3.3 Optimal approximate counting

Recall from Section 1.2 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 we propose here is entirely analogous to the *exact* counting algorithm of Brassard *et al.* [5, 14, 6], and we give only a sketch of it. The algorithm first invokes the procedure $C(X, P)$ of Theorem 3.1 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 1.10 on the expected number of queries made by the algorithm.

Acknowledgements

We would like to thank Andris Ambainis for showing us how to optimize the algorithm of Corollary 1.9 for computing the kth-smallest element, Lov Grover for stimulating discussions, Michele Mosca for sending us a copy of [14] and explaining the details of the exact counting algorithm therein, and Umesh Vazirani for his guidance and useful suggestions.

References

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

[2] 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.

[3] 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.

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

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

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

[7] H. Buhrman, R. Cleve and A. Wigderson. Quantum vs. classical communication and computation. *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 1998, pp. 63–68.

[8] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. Quantum Physics e-Print archive, `http://xxx.lanl.gov/abs/quant-ph/9607014`, 1996.

[9] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser. A limit on the speed of quantum computation in determining parity. Quantum Physics e-Print archive, `http://xxx.lanl.gov/abs/quant-ph/9802045`, 1998.

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

[11] L.K. Grover. A fast quantum mechanical algorithm for estimating the median. Quantum Physics e-Print archive, http://xxx.lanl.gov/abs/quant-ph/9607024, 1996.

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

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

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

[15] 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.

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

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

[18] U. Vazirani. Personal communication, 1997.

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

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

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 [13] says that there is a succint representation for p^{sym} as a *univariate* polynomial.

Fact A.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 *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 $|T_d(1+x)| \le e^{2\sqrt{2x+x^2}}$, for $x \ge 0$.

Fact A.3 *Let p be a polynomial of degree at most d. Then, for $|x| > 1$,*

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

A proof of this fact may be found in Section 2.7 of [17]. We require an easy corollary of this fact.

Corollary A.4 *Let p be a polynomial of degree at most d, with $|p(x)| \le c$ for $|x| \le a$, for some $a > 0$. Then, for all $|x| \ge a$,*

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

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 [16] and Section 2.7 of [17].

Fact A.5 *Let p be a polynomial of degree d. Then, for $x \in [-1, 1]$,*

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

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 $\|t\| = \max_{-\pi \le x \le \pi} |t(x)|$.

Fact A.6 *Let t be a trigonometric polynomial of degree d. Then, for $x \in [-\pi, \pi]$,*

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