Lower Bounds for Quantum Search and Derandomization

Harry Buhrman[*] Ronald de Wolf[†]

November 18, 1998

Abstract

We prove lower bounds on the error probability of a quantum algorithm for searching through an unordered list of N items, as a function of the number T of queries it makes. In particular, if $T \in O(\sqrt{N})$ then the error is lower bounded by a constant. If we want error $\leq 1/2^N$ then we need $T \in \Omega(N)$ queries. We apply this to show that a quantum computer cannot do much better than a classical computer when amplifying the success probability of an RP-machine. A classical computer can achieve error $\leq 1/2^k$ using k applications of the RP-machine, a quantum computer still needs at least ck applications for this (when treating the machine as a black-box), where $c > 0$ is a constant independent of k. Furthermore, we prove a lower bound of $\Omega(\sqrt{\log N}/\log\log N)$ queries for quantum bounded-error search of an *ordered* list of N items.

1 Introduction

Suppose we have an unsorted list of N items and we want to find an item with some specific property. For instance we want to find an item with a specific value at one of its fields. In the worst case, a classical deterministic or randomized computer will have to look at $\Theta(N)$ items to have a high probability of finding such an item if there is one. On the other hand, Grover's quantum search algorithm can perform look-ups or queries in superposition, and finds the desired item with high probability using only $O(\sqrt{N})$ queries. The following is known about the error probability ε in quantum search:

- ε can be made an arbitrarily small constant using $O(\sqrt{N})$ queries [Gro96] but not using $o(\sqrt{N})$ queries [BBBV97, BBHT98, Zal97, BBC+98, Gro98a].
- ε can be made $\leq 1/2^{N^\alpha}$ using $O(N^{0.5+\alpha})$ queries [BCW98, Theorem 1.16].
- If we want no error at all ($\varepsilon = 0$), then we need N queries [BBC+98, Corollary 6.2].

Many applications of quantum computing will need to apply quantum search several times as a subroutine. We should avoid that the errors of each application add up to an overall error that is too big. Accordingly, we should make the error probability of each application as small as possible, if necessary by spending slightly more than $O(\sqrt{N})$ queries.

We give a detailed analysis of the trade-off between the error probability of a quantum search algorithm and the number of queries it uses. We obtain the following lower bound on ε in terms of the number T of queries that the algorithm uses:

$$\varepsilon \in \Omega\left(e^{-4bT^2/N - 8T/\sqrt{N}}\right),$$

[*] CWI, P.O. Box 94709, Amsterdam, The Netherlands. E-mail: buhrman@cwi.nl.

[†] CWI and University of Amsterdam. E-mail: rdewolf@cwi.nl.

where b is some fixed constant and we assume $T < N$. Our proof first translates a quantum search algorithm with T queries to a multivariate polynomial of degree $d \leq 2T$ that has certain properties, and then uses techniques from [Pat92] and [CR92] to prove a lower bound on ε in terms of d. This implies a lower bound in terms of T.[1] In particular, this bound implies that ε cannot be made $o(1)$ using only $O(\sqrt{N})$ queries. Also, ε can only be made $\leq 1/2^N$ using $\Omega(N)$ queries.

In Section 5 we apply this bound to the derandomization of classical RP-machines (RP is the class of languages that can be recognized in polynomial time with one-sided error at most 1/2). A classical computer can achieve error $\leq 1/2^k$ by running the RP-machine k times and answering 'yes' iff at least one of those k runs answered 'yes'. Since this is basically a search among k items, we would expect a quantum computer to be able to achieve error $\leq 1/2^k$ using roughly $\sqrt{k}$ applications of the RP-machine. Somewhat surprisingly, we show that a quantum computer can*not* do much better than the classical computer: it would also need at least ck applications of the machine to obtain error $\leq 1/2^k$ (when treating the machine as a black-box). Here $c > 0$ does not depend on k. We interpret this as follows: general results on amplitude amplification [BHT98, Gro98b, Mos98] show that a quantum computer can achieve a *square-root* speed-up when amplifying a very small success probability to a constant one, but our result shows that it can achieve at most a *linear* speed-up when amplifying a constant success probability to a probability very close to 1.

Finally, in Section 6 we look at the problem of searching an *ordered* list of N items (ordered according to some key field of the items). Since many databases in practice are ordered rather than unordered, we feel this problem merits as much attention as the unordered search has received so far in the quantum computing literature. Classically, we can search such an ordered list with only $\log N$ queries using binary search. It is unknown whether a quantum computer can improve on this. However, we show that it cannot improve much more than a square-root: we prove a lower bound of $\Omega(\sqrt{\log N}/\log\log N)$ queries for bounded-error quantum search in this setting, using a novel kind of quantum reduction from the PARITY-problem.

To summarize:

- We prove a general lower bound on the error in quantum search of an unordered list.
- We apply this to show that a quantum computer can achieve at most a linear speed-up when amplifying an already-big success probability.
- We prove a lower bound of roughly $\sqrt{\log N}$ for quantum search of an ordered list of N items.

2 Preliminaries

In this section we define the setting of quantum gate networks (which are equivalent to quantum Turing machines [Yao93]) and queries.

A *qubit* is a superposition $\alpha_0|0\rangle + \alpha_1|1\rangle$ of both values of a classical bit. Similarly, a register of m qubits is a superposition $|\phi\rangle$ of all 2^m classical bitstrings of m bits, written

$$|\phi\rangle = \sum_{k\in\{0,1\}^m} \alpha_k|k\rangle.$$

Here α_k is a complex number, called the *amplitude* of state $|k\rangle$. If we observe $|\phi\rangle$ we will see one and only one $|k\rangle$. The probability of seeing one specific $|k\rangle$ is given by $|\alpha_k|^2$. Hence we must have $\sum_{k\in\{0,1\}^m}|\alpha_k|^2 = 1$. After observing $|\phi\rangle$ and seeing $|k\rangle$, the superposition $|\phi\rangle$ has collapsed to $|k\rangle$.

[1] Nayak and Wu [NW98] also use polynomial-techniques from [BBC$^+$98] and [Pat92], in order to prove lower bounds for quantum computing the median and mean of a function.

If we do not observe a state, quantum mechanics tells us that it will evolve unitarily. This means that the vector of amplitudes is transformed according to a linear operator that preserves norm (so the sum of the amplitudes squared remains 1). A unitary operator U always has an inverse U^{-1}, which equals its conjugate transpose U^*. A quantum gate network working on m qubits is like a classical circuit working on m classical bits, except that instead of AND-, OR-, and NOT-gates we have quantum gates which operate unitarily on one or more qubits. A quantum gate network transforms an initial state into a final state much in the way a classical circuit transforms its input into one or more output bits. It is known that operations on one or two qubits at a time are sufficient to build any unitary transformation [BBC$^+$95]. The most common measure of complexity of a quantum gate network is the number of elementary quantum gates it contains, but in this paper we will disregard this and only count the number of queries.

Making queries to a list $X = (x_0, \ldots, x_{N-1})$ of N bits is incorporated in the model as follows. Classically, making a query to X means inputting some $j \in \{0, \ldots, N-1\}$ into a black-box, and receiving the value x_j as output. A query gate O (for "oracle") performs the corresponding mapping, which is our only way to access the bits x_j:

$$|j, 0, \overline{0}\rangle \rightarrow |j, x_j, \overline{0}\rangle,$$

where $\overline{0}$ is a string of zeroes. Because O must be reversible, it also maps

$$|j, 1, \overline{0}\rangle \rightarrow |j, \overline{x_j}, \overline{0}\rangle.$$

We will look at quantum networks that contain both elementary gates and query gates, but only count the latter. The advantage of a quantum computer over a classical computer is its ability to make queries in superposition: applying O once to the state $\frac{1}{\sqrt{N}} \sum_j |j, 0, \overline{0}\rangle$ results in $\frac{1}{\sqrt{N}} \sum_j |j, x_j, \overline{0}\rangle$, which in some sense "contains" all the bits x_j.

In terms of linear algebra, a quantum gate network A with T queries can be viewed as follows: first A applies some unitary operation U_0 to the initial state, then it applies O, then it applies another U_1, another O, and so on up till U_T. Thus A corresponds to a big unitary transformation

$$A = U_T O U_{T-1} O \ldots O U_1 O U_0.$$

The behavior of O depends on X, but the U_i are fixed unitary transformations independent of X. We fix the initial state to $|\overline{0}\rangle$, independent of X. The final state is then a superposition $A|\overline{0}\rangle$ which depends on X only via the T query gates.

One specific bit of the final state (the rightmost one, say) is considered the output bit. The output of the network is defined as the value we obtain if we observe this bit. Note that the output is a random variable. The acceptance probability of a quantum network on a specific black-box X is defined to be the probability that the output is 1. The key lemma of [BBC$^+$98] gives the following relation between a T-query network and a polynomial that expresses its acceptance probability as a function of X (such a relation is also implicit in some of the proofs of [FR98, FFKL93]):

Lemma 1 *The acceptance probability of a quantum network that makes T queries to a black-box X, can be written as a real-valued multilinear N-variate polynomial $P(X)$ of degree at most $2T$.*

Note that if we want to compute a Boolean function, then the acceptance probability $P(X)$ should be close to 1 if $f(X) = 1$, and $P(X)$ should be close to 0 if $f(X) = 0$. Since the degree of P is $\leq 2T$, a lower bound on the degree of a polynomial with such properties implies a lower bound on T. See [BBC$^+$98] for some of the lower bounds on quantum query complexity that can be obtained in this way.

An N-variate polynomial P of degree d can be reduced to a single-variate one in the following way (due to [MP68]). Let P^{sym} be the polynomial that averages P over all permutations of its input:

$$P^{sym}(X) = \frac{\sum_{\pi \in S_N} P(\pi(X))}{N!}.$$

P^{sym} is an N-variate polynomial of degree at most d. It can be shown that there is a single-variate polynomial Q of degree at most d, such that $P^{sym}(X) = Q(|X|)$ for all $X \in \{0,1\}^N$. Here $|X|$ denotes the Hamming weight (number of 1s) of X.

3 Lower Bound on the Error in Quantum Search

In this section we establish a general lower bound on the error probability in quantum search. Consider an unordered list of N items. We will abstract from the specific contents of the items, treating the list like a kind of black-box. A query at place j of the list just returns one bit x_j, indicating whether the jth item on the list has the property we are looking for. A query gate performs the following mapping, which is our only access to the bits x_j:

$$|j, b, \overline{0}\rangle \rightarrow |j, b \oplus x_j, \overline{0}\rangle,$$

where b is a bit and $\overline{0}$ is a string of zeroes. The aim is to find a j such that $x_j = 1$ (if there is one), using as few queries as possible.

Rather than proving a lower bound on search directly, we will prove a lower bound on computing the OR-function (i.e. determining whether X contains at least one 1). This clearly reduces to search. The main idea of our proof is the following. By the lemma of the previous section, the acceptance probability of a quantum computer with T queries that computes the OR with error probability $\leq \varepsilon$ can be written as a multivariate polynomial of degree $\leq 2T$ of the N bits in the list. This polynomial can be reduced to a single-variate polynomial s of degree $d \leq 2T$ with the following properties:

$s(0) = 0$ [2]
$1 - \varepsilon \leq s(x) \leq 1$ for all integers $x \in [1, N]$

We will prove a lower bound on ε in terms of d, which implies a lower bound in terms of T. Because we can achieve $\varepsilon = 0$ iff $T = N$ [BBC$^+$98, Proposition 6.1], we assume $T < N$ and hence $\varepsilon > 0$.

Define $p(x) = 1 - s(N - x)$. Then p has degree d and

$0 \leq p(x) \leq \varepsilon$ for all integers $x \in [0, N-1]$
$p(N) = 1$

Thus p is "small" at integer points in $[0, N-1]$ and "big" at N. Coppersmith and Rivlin [CR92, p. 980] prove the following theorem, which allows us to show that p is also "small" at *non*-integer points in $[0, N-1]$.

[2] Since we can always test whether we actually found a solution at the expense of one more query, we can assume the algorithm always gives the right answer 'no' if the list contains only 0s. Hence $s(0) = 0$. However, our results remain unaffected if we allow a small error here also (i.e. $0 \leq s(0) \leq \varepsilon$).

Theorem 1 (Coppersmith & Rivlin) *There exist positive constants a and b with the following property. For every polynomial p of degree d such that*

$$|p(x)| \leq 1 \textit{ for all integers } x \in [0, n]$$

and any $\delta > 0$ such that $n \geq \delta d^2$, we have

$$|p(x)| < ae^{b/\delta} \textit{ for all real } x \in [0, n].$$

Let $\delta = (N-1)/d^2$. Applying Coppersmith and Rivlin's theorem to p/ε (which is bounded by 1 at integer points) we obtain:

$$|p(x)| < \varepsilon a e^{b/\delta} \text{ for all real } x \in [0, N-1].$$

Now we rescale p to $q(x) = p((x+1)(N-1)/2)$ (i.e. the domain $[0, N-1]$ is transformed to $[-1, 1]$), which has the following properties:

$|q(x)| < \varepsilon a e^{b/\delta}$ for all real $x \in [-1, 1]$
For $\mu = 2/(N-1)$ we have $q(1+\mu) = p(N) = 1$

Thus q is "small" on all $x \in [-1, 1]$ and "big" just outside this interval ($q(1+\mu) = 1$).

Let T_d denote the degree-d Chebyshev polynomial [Riv90]:

$$T_d(x) = \frac{1}{2}\left(\left(x + \sqrt{x^2-1}\right)^d + \left(x - \sqrt{x^2-1}\right)^d\right).$$

The following is known:

- If q is a polynomial of degree d such that $|q(x)| \leq c$ for all $x \in [-1, 1]$ then $|q(x)| \leq c|T_d(x)|$ for all $|x| \geq 1$ [Pat92, Fact 2][Riv90, p.108]
- $T_d(1+\mu) \leq e^{2d\sqrt{2\mu+\mu^2}}$ for all $\mu \geq 0$ [Pat92, p.471, before Fact 2][3]

Linking all this we obtain

$$1 = q(1+\mu) \leq \varepsilon a e^{b/\delta} |T_d(1+\mu)| \leq \varepsilon a e^{b/\delta + 2d\sqrt{2\mu+\mu^2}}.$$

This shows that if q is "big" just outside the interval $[-1, 1]$, then it cannot have been very small inside this interval, so ε cannot have been very small. Substituting $\delta = (N-1)/d^2$ and $\mu = 2/(N-1)$ we obtain the following lower bound on ε:

$$\varepsilon \geq \frac{1}{a} e^{-bd^2/(N-1) - 4d/\sqrt{N/(N-1)^2}}.$$

Since $d \leq 2T$, where T is the number of queries of the quantum search algorithm, we have (simplifying a bit):

Theorem 2 *If $T < N$ then $\varepsilon \in \Omega\left(e^{-4bT^2/N - 8T/\sqrt{N}}\right)$.*

We note some special cases of this general theorem:

[3] For $x = 1+\mu$: $T_d(x) \leq (x + \sqrt{x^2-1})^d = (1 + \mu + \sqrt{2\mu+\mu^2})^d \leq (1 + 2\sqrt{2\mu+\mu^2})^d \leq e^{2d\sqrt{2\mu+\mu^2}}$ (Paturi, personal communication).

Corollary 1 *No quantum network for bounded-error search of an unordered list that uses $O(\sqrt{N})$ queries can have error probability $o(1)$.*

For instance, an error $\leq 1/N$ cannot be achieved using only $O(\sqrt{N})$ queries.[4]

Corollary 2 *Every quantum network for bounded-error search of an unordered list that uses $\leq N^{0.5+\alpha}$ queries ($\alpha \geq 0$) must have error probability $\Omega\left(1/2^{cN^{2\alpha}}\right)$ (where $c > 0$ is some fixed constant).*

In particular, this shows that we cannot obtain error probability $\leq 1/2^N$ unless we have $\alpha = 0.5$ and thus use $\Omega(N)$ queries. [BCW98, Theorem 1.16] proves the upper bound that the error probability can be made as small as $1/2^{N^\alpha}$ using $O(N^{0.5+\alpha})$ queries, so there is still a gap between upper and lower bound.

Finally, a lower bound on T in terms of ε and N:

Corollary 3 *If $T(N)/\sqrt{N} \to \infty$ but $T < N$, then $T \in \Omega\left(\sqrt{N \log(1/\varepsilon)}\right)$.*

4 The Influence of the Number of Solutions

Suppose we have a quantum search algorithm that uses T queries and works well (i.e. has error $\leq \varepsilon$) whenever the number of 1s in the list of N items is either 0 or at least t. (Here t is some fixed number $< N$.) Such an algorithm induces a polynomial of degree $d \leq 2T$ with the following properties:

$s(0) = 0$
$1 - \varepsilon \leq s(x) \leq 1$ for all integers $x \in [t, N]$

Define $p(x) = 1 - s(N - x)$, which has degree d and

$0 \leq p(x) \leq \varepsilon$ for all integers $x \in [0, N - t]$
$p(N) = 1$

Now we define $q(x) = p((x+1)(N-t)/2)$, $\delta = (N-t)/d^2$ and $\mu = 2t/(N-t)$, and derive completely analogous to the previous section:

$$\begin{aligned} 1 &= q(1+\mu) \leq \varepsilon a e^{b/\delta} |T_d(1+\mu)| \leq \varepsilon a e^{b/\delta + 2d\sqrt{2\mu+\mu^2}} \\ &= \varepsilon a e^{bd^2/(N-t) + 2d\sqrt{4t/(N-t) + 4t^2/(N-t)^2}} \\ &= \varepsilon a e^{bd^2/(N-t) + 4d\sqrt{tN/(N-t)^2}}. \end{aligned}$$

Hence for quantum search in this situation we have the bound:

$$\varepsilon \in \Omega\left(e^{-bd^2/(N-t) - 4d\sqrt{tN/(N-t)^2}}\right) \in \Omega\left(e^{-4bT^2/(N-t) - 8T\sqrt{tN/(N-t)^2}}\right).$$

[BBHT98] proves that an expected number of $O(\sqrt{N/t})$ queries is sufficient to search with high probability. If we put $T = c\sqrt{N/t}$ then the lower bound on the error probability becomes roughly

[4] Which is too bad, because such a small error would reduce the quantum complexity of Σ_2 (the second level of the polynomial hierarchy) from $O(\sqrt{2^n}n)$ to $O(\sqrt{2^n})$ [BCW98].

$\Omega(e^{-c'c})$ (for some constant $c' > 0$), which can indeed be made arbitrarily small by increasing c. On the other hand, if $T \in o(\sqrt{N/t})$ then the lower bound on the error goes to the constant $1/a$ for $N \to \infty$ and $t = o(N)$. Now if we were able to achieve some error $< 1/2$ using $o(\sqrt{N/t})$ queries, we could also make the error $< 1/a$ by repeating a constant number of times, which would still take only $o(\sqrt{N/t})$ queries. This shows that we can*not* achieve error $< 1/2$ using $o(\sqrt{N/t})$ queries. Thus the $O(\sqrt{N/t})$ upper bound is tight up to a constant factor (as already shown in a different way in [BBHT98]).

5 Application to Derandomization of RP

Let A be some RP-algorithm for a language L with running time $\leq p(n)$. A always gives the right answer 'no' for every input $x \notin L$, and gives the right answer 'yes' with probability at least 1/2 for every $x \in L$. We want to lower the error probability using as few calls to A as possible. For a fixed input x of length n we can consider A as a black-box of $N \leq 2^{p(n)}$ items. Each item corresponds to the value A outputs when given a specific random string (A can use at most $p(n)$ random bits and hence at most $2^{p(n)}$ distinct random strings). By definition of RP, this black-box satisfies the promise that either it contains 0 1s (if $x \notin L$) or at least $N/2$ 1s (if $x \in L$).

A classical computer can improve the error probability to at most $1/2^k$ by making k black-box queries (i.e. k applications of the algorithm on k different random strings) and answering 'yes' iff at least one those k queries answered 'yes'. How much better can a quantum computer do, if we only allow it to call A as a black-box? Note that the classical method basically searches through a list of k items, looking for a 1. Accordingly, the following quantum algorithm suggests itself: select k random strings and search whether one of these gives a 'yes' in $O(\sqrt{k})$ applications of the algorithm. Thus we would expect a quantum computer to be able to achieve the same error probability $\leq 1/2^k$ using roughly $\sqrt{k}$ applications of the algorithm instead of k.

However, note that the situation here corresponds exactly to the previous section with $t = N/2$. Thus if the quantum computer makes T queries and has error probability ε on the worst-case black-box, then

$$\varepsilon \in \Omega\left(e^{-8bT^2/N - 8T\sqrt{2}}\right).$$

If we want $\varepsilon \leq 1/2^k$ (for some fixed k and all N), it follows that $T \geq ck$, for some $c > 0$ that does not depend on k.[5] Thus the quantum algorithm can*not* achieve the square-root speed-up that we expected; it can achieve at most a linear speed-up.

Why does the above-mentioned $\sqrt{k}$-method not work? The reason is that the quantum searching algorithm itself has some error probability, in addition to the probability $\leq 1/2^k$ that the chosen sample of k items does not contain a 1 when the larger list of N items *does* contain a 1. The error introduced by quantum search can only be made sufficiently small at the cost of increasing k and/or the number of queries spent.

In sum: on a classical computer we can amplify an RP-algorithm to error probability $\varepsilon \leq 1/2^k$ using k applications of the algorithm, on a quantum computer we cannot do much better: we still need at least ck applications to achieve error $\varepsilon \leq 1/2^k$, provided we use the RP-machine only as a black-box.

[5] For sufficiently large k, c will be roughly $1/8\sqrt{2}\log e \approx 0.06$.

6 Lower Bound on Search in an Ordered List

Grover's algorithm can find a specific item in an unordered list of N items with high probability, using only $O(\sqrt{N})$ queries (a.k.a. database look-ups), whereas a classical algorithm needs $\Theta(N)$ queries for this. There exist several lower-bound proofs that show that the $O(\sqrt{N})$ is optimal [BBBV97, BBHT98, Zal97, BBC$^+$98, Gro98a].

What about search in a list of N items which is *ordered* according to some key-value of each item? A classical deterministic algorithm can search such a list using $\log N$ queries by means of binary search (each query can effectively halve the relevant part of the list: looking at the key of the middle item of the list tells you whether the item you are searching for is in the first or the second half of the list). How much better can we do on a quantum computer? Can we again get a square-root speed-up? Here we show that the speed-up cannot be much better than a square-root: we prove a lower bound of $\Omega(\sqrt{\log N}/\log\log N)$ queries for bounded-error quantum search of an ordered list. In contrast, we have no upper bound better than the classical $\log N$.

We will formalize a query on an ordered list as follows, abstracting from the specific contents of the key field. The list is viewed as a list of N bits, $x_0, \ldots, x_{N-1}$, and there is an unknown number i such that $x_j = 1$ iff $j \leq i$. Here x_j being 1 can be interpreted as saying that the jth item on the list has a key-value smaller or equal to the value we are looking for. The goal is to find the number i, which is the point in the list where the looked-for item resides, using as few queries as possible. In quantum network terms, a query corresponds to a gate C that maps

$$|j, b, \overline{0}\rangle \rightarrow |j, b \oplus x_j, \overline{0}\rangle.$$

The following theorem proves a lower bound of roughly $\sqrt{\log N}$ queries for quantum searching an ordered list with bounded error probability. To improve readability, we have deferred some of the more technical details to the appendix. Basically these show that we can approximately simulate the gate C using roughly $\sqrt{\log N}$ queries to a black-box of $\log N$ bits that represents the number i.

Theorem 3 *A quantum network for bounded-error search of an ordered list of N items must use at least $\Omega(\sqrt{\log N}/\log\log N)$ queries.*

Proof Suppose we have a network S for bounded-error ordered search that uses T queries to find the number i hidden in an ordered black-box X with high probability. Since $\log N$ queries are sufficient for this (classical binary search), we can assume $T \leq \log N$. We will show how we can get from S to a network $\widetilde{S}$ that determines the whole contents of an arbitrary black-box Y of $\log N$ bits with high probability, using only $T \cdot O(\sqrt{\log N}\log\log N)$ queries to Y. This would allow us to compute the PARITY-function of Y (i.e. whether or not Y contains odd many 1s). Since we have a $(\log N)/2$ lower bound for the latter [BBC$^+$98, Proposition 6.4], we have

$$T \cdot O(\sqrt{\log N}\log\log N) \geq \frac{\log N}{2},$$

from which the theorem follows.

So let Y be an arbitrary black-box of $\log N$ bits. This represents a number $i \in \{0, \ldots, N-1\}$. Let $X = (x_0, \ldots, x_{N-1})$ be the ordered black-box corresponding to i, so $x_j = 1$ iff $j \leq i$. The network S, when allowed to make queries to X, outputs the number i with high probability. A query-gate C for X maps

$$|j, b, \overline{0}\rangle \rightarrow |j, b \oplus x_j, \overline{0}\rangle.$$

Since $x_j = 1$ iff $j \leq i$, Lemmas 2 and 3 of the appendix imply that there is a quantum network $\widetilde{C}$ that uses $O(\sqrt{\log N} \log\log N)$ queries to Y and maps

$$|j, b, \overline{0}\rangle \rightarrow |j, b \oplus x_j, \overline{0}\rangle + |j\rangle|W_{jb}\rangle,$$

where $\| \, |W_{jb}\rangle \, \| \leq \eta/\log N$ for all j, b, for some small fixed η of our choice.

Let $\widetilde{S}$ be obtained from S by replacing all T C-gates by $\widetilde{C}$-networks. Note that $\widetilde{S}$ contains $T \cdot O(\sqrt{\log N} \log\log N)$ queries to Y. Consider the way $\widetilde{S}$ acts on initial state $|\overline{0}\rangle$, compared to S. Each replacement of C by $\widetilde{C}$ introduces an error, but each of these errors is at most $\sqrt{2}\eta/\log N$ in Euclidean norm by Lemma 4. By unitarity these T errors add linearly, so the final states will be close together:

$$\| \, S|\overline{0}\rangle - \widetilde{S}|\overline{0}\rangle \, \| \leq T\sqrt{2}\eta/\log N \leq \sqrt{2}\eta.$$

Since observing the final state $S|\overline{0}\rangle$ yields the number i with high probability, observing $\widetilde{S}|\overline{0}\rangle$ will also yield i with high probability. Thus the network $\widetilde{S}$ allows us to learn i, and hence the whole black-box Y. □

Acknowledgements

We would like to thank David Deutsch, Wim van Dam and Mike Mosca for discussions which emphasized the importance of making the error in quantum search as small as possible.

References

[BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. quant-ph/9701001.

[BBC$^+$95] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995.

[BBC$^+$98] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. In *Proceedings of 39th FOCS*, pages 352–361, 1998. also quant-ph/9802049.

[BBHT98] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998. Earlier version in Physcomp'96; also quant-ph/9605034.

[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation (preliminary version). In *Proceedings of 30th STOC*, pages 63–68, 1998. quant-ph/9802040.

[BHT98] G. Brassard, P. Høyer, and A. Tapp. Quantum counting. In *Proceedings of 25th ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 820–831. Springer, 1998. quant-ph/9805082.

[CDNT97] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. quant-ph/9708019, 10 Aug 1997.

[CR92] D. Coppersmith and T. J. Rivlin. The growth of polynomials bounded at equally spaced points. *SIAM Journal on Mathematical Analysis*, 23(4):970–983, 1992.

[DH96] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. quant-ph/9607014, 18 Jul 1996.

[FFKL93] S. Fenner, L. Fortnow, S. Kurtz, and L. Li. An oracle builder's toolkit. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference*, pages 120–131, 1993.

[FR98] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. In *Proceedings of the 13th IEEE Conference on Computational Complexity*, pages 202–209, 1998.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th STOC*, pages 212–219, 1996. quant-ph/9605043.

[Gro98a] L. K. Grover. How fast can a quantum computer search? quant-ph/9809029, 10 Sep 1998.

[Gro98b] L. K. Grover. A framework for fast quantum mechanical algorithms. In *Proceedings of 30th STOC*, pages 53–62, 1998. quant-ph/9711043.

[Mos98] M. Mosca. Quantum searching, counting and amplitude amplification by eigenvector analysis. In *MFCS'98 workshop on Randomized Algorithms*, 1998.

[MP68] M. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 1968. Second, expanded edition 1988.

[NW98] A. Nayak and F. Wu. On the quantum black-box complexity of approximating the mean and the median. quant-ph/9804066, 29 Apr 1998.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions (preliminary version). In *Proceedings of 24th STOC*, pages 468–474, 1992.

[Riv90] T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory.* Wiley-Interscience, second edition, 1990.

[Yao93] A. C-C. Yao. Quantum circuit complexity. In *Proceedings of 34th FOCS*, pages 352–360, 1993.

[Zal97] C. Zalka. Grover's quantum searching algorithm is optimal. quant-ph/9711070, 26 Nov 1997.

A Some Technical Lemmas

Our lower-bound proof for ordered search uses three technical lemmas.

The first lemma can be obtained from the result of Dürr and Høyer [DH96] that a quantum algorithm can find the *minimum* element on a list of N items using $O(\sqrt{N})$ queries. We can use this to find the leftmost bit where two lists differ, which tells us which of the two numbers represented by the two lists is bigger.

Lemma 2 *There exists a quantum algorithm A that with bounded error probability outputs on input j ($0 \leq j \leq N-1$) whether j is smaller or equal to a number i represented by a black-box of $\log N$ bits, using $O(\sqrt{\log N})$ queries to the black-box.*

By standard techniques, we can make the error probability $O(1/\log N)$ by repeating the algorithm $O(\log \log N)$ times.

The second lemma shows how to obtain an approximately "clean" computation that uses no measurements (the proof is as in [CDNT97, Section 3] and [BCW98, Theorem 1.14]).

Lemma 3 *Suppose there exists a quantum algorithm A that uses T queries and outputs a bit x_j with error probability $\leq \varepsilon$ on initial state $|j, \overline{0}\rangle$, for every j, and does not change the j-register. Then there exists a quantum algorithm A' that uses $2T$ queries and no measurements, and maps*

$$|j, b, \overline{0}\rangle \rightarrow |j, b \oplus x_j, \overline{0}\rangle + |j\rangle|W_{jb}\rangle,$$

where $\| \, |W_{jb}\rangle \, \| \leq \sqrt{2\varepsilon}$, for every j and $b \in \{0,1\}$.

Proof The idea is the familiar "compute, copy answer, uncompute"-sequence. By standard techniques, we can assume A itself uses no measurements and is followed by a single measurement. Then there exist amplitudes α_0 and α_1 and unit-length vectors $|V_0\rangle$ and $|V_1\rangle$ such that

$$A|j, 0, \overline{0}\rangle = \alpha_0|j, x_j\rangle|V_0\rangle + \alpha_1|j, \overline{x_j}\rangle|V_1\rangle,$$

and $|\alpha_1|^2 \leq \varepsilon$. For ease of notation, we assume this state is preceded by the bit b. Applying the controlled-not operation that maps $|b, j, x\rangle \rightarrow |b \oplus x, j, x\rangle$, we get

$$\alpha_0|b \oplus x_j, j, x_j\rangle|V_0\rangle + \alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle =$$
$$|b \oplus x_j\rangle \left(\alpha_0|j, x_j\rangle|V_0\rangle + \alpha_1|j, \overline{x_j}\rangle|V_1\rangle\right) + \alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle.$$

Applying $I \otimes A^{-1}$ gives

$$|b \oplus x_j\rangle|j, 0, \overline{0}\rangle + (I \otimes A^{-1}) \left(\alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle\right).$$

Applying an operation B which swaps the first bit and j, we get

$$|j, b \oplus x_j, 0, \overline{0}\rangle + B(I \otimes A^{-1}) \left(\alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle\right).$$

Note that $B(I \otimes A^{-1}) \left(\alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle\right) = |j\rangle|W_{jb}\rangle$ for some $|W_{jb}\rangle$, because A and hence also A^{-1} do not change j. Now

$$\begin{aligned}
\| \, |W_{jb}\rangle \, \| &= \| \, |j\rangle|W_{jb}\rangle \, \| \\
&= \| \, B(I \otimes A^{-1}) \left(\alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle\right) \| \\
&= \| \, \alpha_1|b \oplus \overline{x_j}, j, \overline{x_j}\rangle|V_1\rangle - \alpha_1|b \oplus x_j, j, \overline{x_j}\rangle|V_1\rangle \, \| \\
&= \sqrt{2|\alpha_1|^2} \leq \sqrt{2\varepsilon}.
\end{aligned}$$

Thus the quantum algorithm A' which first applies A, then XORs the answer-bit into b, and then applies A^{-1}, satisfies the lemma. □

The next lemma uses an idea from [CDNT97]. It shows that if we can simulate a gate C by means of a network $\widetilde{C}$ that works well on basis states, then $\widetilde{C}$ also works well on superpositions of basis states.

Lemma 4 *Let C and $\widetilde{C}$ be unitary transformations such that*

$$\begin{array}{l} C : |j, b, \overline{0}\rangle \to |j, b \oplus x_j, \overline{0}\rangle \\ \widetilde{C} : |j, b, \overline{0}\rangle \to |j, b \oplus x_j, \overline{0}\rangle + |j\rangle|W_{jb}\rangle \end{array}$$

If $\parallel |W_{jb}\rangle \parallel \leq \varepsilon$ for every $j \in \{0, \ldots, N-1\}$ and $b \in \{0, 1\}$, and $|\phi\rangle = \sum_{j,b} \alpha_{jb}|j, b, \overline{0}\rangle$ has norm 1, then

$$\parallel C|\phi\rangle - \tilde{C}|\phi\rangle \parallel \leq \sqrt{2}\varepsilon.$$

Proof

$$\begin{array}{rcl} \parallel C|\phi\rangle - \tilde{C}|\phi\rangle \parallel & = & \parallel \sum\limits_{j,b} \alpha_{jb}|j\rangle|W_{jb}\rangle \parallel \\ & \leq & \parallel \sum\limits_{j} \alpha_{j0}|j\rangle|W_{j0}\rangle \parallel + \parallel \sum\limits_{j} \alpha_{j1}|j\rangle|W_{j1}\rangle \parallel \\ & \overset{(1)}{=} & \sqrt{\sum\limits_{j} |\alpha_{j0}|^2 \parallel |j\rangle|W_{j0}\rangle \parallel^2} + \sqrt{\sum\limits_{j} |\alpha_{j1}|^2 \parallel |j\rangle|W_{j1}\rangle \parallel^2} \\ & \leq & \varepsilon\sqrt{\sum\limits_{j} |\alpha_{j0}|^2} + \varepsilon\sqrt{\sum\limits_{j} |\alpha_{j1}|^2} \overset{(2)}{\leq} \sqrt{2}\varepsilon. \end{array}$$

Here (1) holds because the states $|j\rangle|W_{jb}\rangle$ in $\sum_j \alpha_{jb}|j\rangle|W_{jb}\rangle$ are all orthogonal, and (2) holds because $\sqrt{a} + \sqrt{1-a} \leq \sqrt{2}$ for all $a \in [0, 1]$. □