A Lower Bound for Quantum Search of an Ordered List

Harry Buhrman* Ronald de Wolf†

June 21, 1999

Abstract

It is known that a quantum computer can search an unordered list of N items using $O(\sqrt{N})$ look-ups, which is quadratically faster than any classical algorithm. We examine the case where the list is ordered, and show that no quantum computer can do with fewer than $\Omega(\sqrt{\log N}/\log\log N)$ look-ups.

1 Introduction

Search is a basic operation in computer science and its complexity for classical computers has been well studied [Knu98]. It is known that a classical randomized algorithm that searches for some specific item in an unordered list of length N has to query at least $N/2$ items of the list in order to have success probability $\geq 2/3$. In contrast, a *quantum* computer can make queries in superposition and can search such a list using only $O(\sqrt{N})$ queries [Gro96]. It is known that the $O(\sqrt{N})$ is optimal [BBBV97, BBHT98, Zal97, BBC$^+$98, Gro98]. If we do not want to allow a small error probability then even a quantum computer needs N queries [BBC$^+$98].

Until recently, not much attention had been paid to the quantum complexity of searching a list which is *ordered* according to some key-field of the items. Classically, we can search such a list with only $\log N$ queries using 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); $\log N$ is also the classical lower bound, even in the bounded-error case. How much better can we do on a quantum computer? We show that a quantum computer cannot improve this 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. The proof shows how searching an ordered list of N items enables us to retrieve the whole contents of an ordered list of $\log N$ bits. For the latter problem a tight bound is known [BBC$^+$98, FGGS98a, vD98].

Our lower bound was the first for quantum ordered search (it first appeared in [BW98]). It has recently been improved by means of a different proof technique to $(\log N)/2\log\log N$ [FGGS98b] and then to $(\log N)/12 - O(1)$ [Amb99]. Thus at most a linear speed-up is possible over classical computers. Such a linear quantum speed-up is indeed possible: an upper bound of $0.53\log N$ can be achieved [FGGS99].

2 Definitions

In this section we briefly define the setting of quantum gate networks and queries.

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

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

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 $|k\rangle$. The (Euclidean) *norm* of $|\phi\rangle$ is $\parallel |\phi\rangle \parallel = \sqrt{\sum_k |\alpha_k|^2}$. The (Euclidean) *distance* between $|\phi\rangle$ and $|\psi\rangle$ is $\parallel |\phi\rangle - |\psi\rangle \parallel$. We use $|\vec{0}\rangle$ to denote the state where all qubits are zero. If b is a bit, $\overline{b}$ denotes its negation.

If we observe or measure $|\phi\rangle$ 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 $|k\rangle$.

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. We will use the term 'quantum algorithm' loosely, to refer to a quantum network or a family of networks for different input sizes.

We 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, $X = (x_0, \ldots, x_{N-1})$, and there is an unknown number i such that $x_j = 1$ iff $j \leq i$. We call i the *step* of X. Here x_j is the result of a comparison, indicating whether 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 that maps

$$|j, b, w\rangle \rightarrow |j, b \oplus x_j, w\rangle.$$

Thus the bit x_j is XORed into some specific bit b of the input; w represents the workspace, which remains unaffected. With some abuse of notation we denote this unitary transformation by X, and sometimes call it a 'black-box'.

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 X, then it applies another unitary operation U_1, another X, and so on up till U_T. Thus A corresponds to a unitary transformation

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

Without loss of generality we fix the initial state to $|\vec{0}\rangle$, independent of X. The U_i are fixed unitary transformations independent of X. The final state is thus a superposition $A|\vec{0}\rangle$ which depends on X only via the T query gates.

3 Intuition

Before plunging into the technicalities of the proof let us briefly sketch the main idea, ignoring the error probabilities for now. Suppose we have a quantum network S that uses T queries to determine the step i of any ordered black-box X of N items. For ease of notation we assume N is a power of 2, so $\log N$ is an integer.

Suppose also that we are given a black-box Y of $\log N$ bits, and we want to determine its contents. We can use S to do this, as follows. The sequence of bits in Y is the binary representation of some number $i \in [0, N-1]$. Define X as the ordered black-box of size N where the step occurs at position i: $x_j = 1$ for $j \leq i$ and $x_j = 0$ for $j > i$. Running S on X would give us i, and hence Y. Unfortunately we do not have the possibility to query X; we can only query Y.

However, we can *simulate* an X-query using Y-queries. An X-query is basically a mapping from a given number j to the bit x_j, where $x_j = 1$ iff $j \leq i$. Both j and i are $\log N$-bit numbers, and the leftmost (= most significant) bit where their binary representations differ determines whether $j \leq i$. Using Grover's algorithm we can *find* this bit using roughly $\sqrt{\log N}$ queries to Y (which holds i), and hence learn x_j. Thus we can simulate an X-query by $\sqrt{\log N}$ Y-queries.

Now if we replace each of the T X-queries in S by such a simulation, we obtain a network with roughly $T \cdot \sqrt{\log N}$ Y-queries that computes i (and hence the whole Y). Knowing Y would enable us for instance to compute the PARITY of Y (i.e. whether the number of 1s in Y is odd), for which a lower bound of $(\log N)/2$ Y-queries is known [BBC$^+$98, FGGS98a]. Hence roughly

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

and the lower bound on T follows. The following technical sections make this idea precise.

4 Simulating Queries to an Ordered Black-Box

Our lower bound proof uses three technical lemmas which we prove first. The task of these lemmas is to show that we can approximately simulate an ordered black-box X with step at i, using roughly $\sqrt{\log N}$ queries to a black-box Y of $\log N$ bits that form the binary representation of i.

Since $x_j = 1$ iff $j \leq i$, we can simulate an X-query if we are able to determine whether $j \leq i$ for given j. By a result of Dürr and Høyer [DH96], there is a bounded-error quantum algorithm that can find the *minimum* element of a list of n items using $O(\sqrt{n})$ queries. We can use this to find the leftmost bit where the binary representations of i and j differ, using $O(\sqrt{\log N})$ Y-queries, thus determining whether $j \leq i$. By standard techniques we can get the error probability down to $\varepsilon = 1/\log N$ by repeating the algorithm $O(\log\log N)$ times. We may assume without loss of generality that this computation does not affect the input j and does not use intermediate measurements. Thus we obtain:

Lemma 1 *There exists a quantum algorithm A that makes $O(\sqrt{\log N}\log\log N)$ queries to a $\log N$-bit black-box Y, such that if Y represents the number i, then for every $j \in [0, N-1]$ A maps*

$$|j, \vec{0}\rangle \rightarrow \alpha|j, x_j\rangle|V_{ij}\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle,$$

where $x_j = 1$ if $j \leq i$ and $x_j = 0$ if $j > i$, $|\beta|^2 \leq \varepsilon = 1/\log N$, and V_{ij} and V'_{ij} are unit-length vectors that depend on i and j.

If we want to simulate an X-query, we must make sure that the simulation does not leave behind used non-zero workspace, since this may destroy interference later on. Thus we must somehow "clean-up" the vector $|V_{ij}\rangle$. The second lemma shows how to obtain an approximately clean computation that uses no measurements (this is by now a standard technique and can be found for instance in [BBBV97, CDNT98, BCW98]).

Lemma 2 *Suppose A is a quantum algorithm that uses T Y-queries and for every $j \in [0, N-1]$ maps*

$$|j, \vec{0}\rangle \to \alpha|j, x_j\rangle|V_{ij}\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle,$$

where $|\beta|^2 \leq \varepsilon$ and V_{ij} and V'_{ij} have unit length.

Then there exists a quantum algorithm A' that uses $2T$ Y-queries and maps

$$|j, b, \vec{0}\rangle \to |j, b \oplus x_j, \vec{0}\rangle + |j\rangle|W_{ijb}\rangle,$$

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

Proof The idea is the familiar "compute, copy answer, uncompute"-sequence. For ease of notation we assume b follows the workspace $\vec{0}$ instead of preceding it. Thus we can write

$$A|j, \vec{0}, b\rangle = \alpha|j, x_j\rangle|V_{ij}\rangle|b\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b\rangle.$$

Applying a controlled-not operation which XORs the answer bit into b, we get

$$\alpha|j, x_j\rangle|V_{ij}\rangle|b \oplus x_j\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle =$$

$$\left(\alpha|j, x_j\rangle|V_{ij}\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle\right)|b \oplus x_j\rangle + \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle - \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus x_j\rangle.$$

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

$$|j, \vec{0}\rangle|b \oplus x_j\rangle + (A^{-1} \otimes I)\left(\beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle - \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus x_j\rangle\right).$$

Because A and hence also A^{-1} do not change j, this superposition can be written as

$$|j, \vec{0}, b \oplus x_j\rangle + |j\rangle|W_{ijb}\rangle,$$

for some vector $|W_{ijb}\rangle$. Now

$$\begin{aligned}
\| \, |W_{ijb}\rangle \, \| &= \| \, |j\rangle|W_{ijb}\rangle \, \| && (1)\\
&= \| \, (A^{-1} \otimes I)\left(\beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle - \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus x_j\rangle\right) \| && (2)\\
&= \| \, \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle - \beta|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus x_j\rangle \, \| && (3)\\
&= \sqrt{|\beta|^2 + |-\beta|^2} && (4)\\
&\leq \sqrt{2\varepsilon}. && (5)
\end{aligned}$$

Here (1) holds because $|j\rangle$ has norm 1. Equality between (2) and (3) holds because $A^{-1} \otimes I$ is unitary and hence preserves norm. Equality between (3) and (4) holds because the two vectors $|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus \overline{x_j}\rangle$ and $|j, \overline{x_j}\rangle|V'_{ij}\rangle|b \oplus x_j\rangle$ in (3) have norm 1 and are orthogonal (they differ in the last bit).

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

We have now shown that we can "cleanly" simulate the operation of black-box X on a basis state $|j, b, \vec{0}\rangle$. It remains to show that the simulation also works well on *superpositions* of basis states. The next lemma proves this, using an idea from [CDNT98].

Lemma 3 *Let X and $\widetilde{X}$ be unitary transformations such that*

$$\begin{aligned} X &: |j, b, \vec{0}\rangle \to |j, b \oplus x_j, \vec{0}\rangle \\ \widetilde{X} &: |j, b, \vec{0}\rangle \to |j, b \oplus x_j, \vec{0}\rangle + |j\rangle|W_{ijb}\rangle \end{aligned}$$

If $\| \, |W_{ijb}\rangle \, \| \leq \varepsilon$ for every i, j, b and $|\phi\rangle = \sum_{j,b} \alpha_{jb}|j, b, \vec{0}\rangle$ has norm 1, then $\| \, X|\phi\rangle - \tilde{X}|\phi\rangle \, \| \leq \varepsilon\sqrt{2}$.

Proof

$$\begin{aligned} \| \, X|\phi\rangle - \tilde{X}|\phi\rangle \, \| &= \| \sum_{j,b} \alpha_{jb}|j\rangle|W_{ijb}\rangle \, \| &(6)\\ &= \| \sum_j \alpha_{j0}|j\rangle|W_{ij0}\rangle + \sum_j \alpha_{j1}|j\rangle|W_{ij1}\rangle \, \| &(7)\\ &\leq \| \sum_j \alpha_{j0}|j\rangle|W_{ij0}\rangle \, \| + \| \sum_j \alpha_{j1}|j\rangle|W_{ij1}\rangle \, \| &(8)\\ &= \sqrt{\sum_j |\alpha_{j0}|^2 \, \| \, |j\rangle|W_{ij0}\rangle \, \|^2} + \sqrt{\sum_j |\alpha_{j1}|^2 \, \| \, |j\rangle|W_{ij1}\rangle \, \|^2} &(9)\\ &\leq \varepsilon \cdot \sqrt{\sum_j |\alpha_{j0}|^2} + \varepsilon \cdot \sqrt{\sum_j |\alpha_{j1}|^2} &(10)\\ &\leq \varepsilon\sqrt{2}. &(11) \end{aligned}$$

The step from (7) to (8) is the triangle inequality. The step from (8) to (9) holds because the states $|j\rangle|W_{ijb}\rangle$ in $\sum_j \alpha_{jb}|j\rangle|W_{ijb}\rangle$ are all orthogonal. The last inequality holds because $\sum_j |\alpha_{j0}|^2 + \sum_j |\alpha_{j1}|^2 = 1$ and $\sqrt{a} + \sqrt{1-a} \leq \sqrt{2}$ for all $a \in [0, 1]$. □

5 Lower Bound for Ordered Search

Theorem 1 *A bounded-error quantum algorithm for 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 bounded-error network S for search that uses T queries to find the step i hidden in an ordered black-box X. 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) with small error probability. Since we have a $(\log N)/2$ lower bound for the latter ([BBC$^+$98, Proposition 6.4] and [FGGS98a]), 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. It represents a number $i \in \{0, \ldots, N-1\}$. Let $X = (x_0, \ldots, x_{N-1})$ be the ordered black-box with step at 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. X maps

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

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

$$|j, b, \vec{0}\rangle \rightarrow |j, b \oplus x_j, \vec{0}\rangle + |j\rangle|W_{ijb}\rangle,$$

where $\| \, |W_{ijb}\rangle \, \| \leq \eta/\log N$ for all i, j, b, for some small fixed η of our choice ($\eta = 0.1$ suffices).

Let $\widetilde{S}$ be obtained from S by replacing all T X-gates by $\widetilde{X}$-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 $|\vec{0}\rangle$, compared to S. Each replacement of X by $\widetilde{X}$ introduces an error, but each of these errors is at most $\eta\sqrt{2}/\log N$ in Euclidean norm by Lemma 3. Using the triangle inequality and the unitarity of the transformations in S and $\widetilde{S}$, it is easy to show that these T errors add at most linearly (see for instance [BBBV97, p.1515]). Hence the final states after S and $\widetilde{S}$ will be close together:

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

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

References

[Amb99] A. Ambainis. A better lower bound for quantum algorithms searching an ordered list. Available at the LANL preprint archive, quant-ph/9902053, 14 Feb 1999.

[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. quant-ph/9503016.

[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. 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. 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.

[BW98] H. Buhrman and R. de Wolf. Lower bounds for quantum search and derandomization. quant-ph/9811046, 18 Nov 1998.

[CDNT98] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Proceedings of 1st NASA QCQC conference*, volume 1509 of *Lecture Notes in Computer Science*. Springer, 1998. quant-ph/9708019.

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

[FGGS98a] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. quant-ph/9802045, 16 Feb 1998.

[FGGS98b] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation for insertion into an ordered list. quant-ph/9812057, 18 Dec 1998.

[FGGS99] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Invariant quantum algorithms for insertion into an ordered list. quant-ph/9901059, 19 Jan 1999.

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

[Gro98] L. K. Grover. How fast can a quantum computer search? quant-ph/9809029, 10 Sep 1998.

[Knu98] D. E. Knuth. *The Art of Computer Programming. Volume 3: Sorting and Searching.* Addison-Wesley, second edition, 1998.

[vD98] W. van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of 39th FOCS*, pages 362–367, 1998. quant-ph/9805006.

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