A Quantum Algorithm for Finding the Minimum

Christoph Dürr* — *Université Paris-Sud*†

Peter Høyer‡ — *Odense University*§

18 July 1996

Let $T[0..N-1]$ be an unsorted table of N items, each holding a value from an ordered set. For simplicity, assume that all values are distinct. The minimum searching problem is to find the index y of an item such that $T[y]$ is minimum. This clearly requires a linear number of probes on a classical probabilistic Turing machine if the table is unsorted.

In this note, we give a simple quantum algorithm which solves the problem using $\mathcal{O}(\sqrt{N})$ probes. The main subroutine is the quantum exponential searching algorithm of [2], which itself is a generalization of Grover's recent quantum searching algorithm [3]. Due to a general lower bound of [1], this is within a constant factor of optimal.

Our algorithm calls this subroutine to find the index of a smaller item than the value determined by a particular *threshold index*. The result is then chosen as the new threshold. This process is repeated until the probability that the item with the threshold index holds the minimum is sufficiently close to one.

If there are $t \geq 1$ such smaller table entries, the subroutine will return one of them with equal probability after an expected number of $\mathcal{O}(\sqrt{N/t})$ iterations. We use the total number of iterations applied as a measure for the running time of the minimum searching algorithm and obtain the following theorem.

Theorem 1 *Let $c \geq 1$ be any constant. By measuring after a total number of $\mathcal{O}(c\sqrt{N})$ iterations, the algorithm given below finds (the index of a item holding) the minimum value with probability at least $1 - \frac{1}{2^c}$.*

We first give the minimum searching algorithm, and then the proof of its optimality.

Quantum Minimum Searching Algorithm

1. Let $c \geq 1$ and m_0 be given.
   Choose threshold index y uniformly at random from $\{0, \ldots, N-1\}$.
2. Repeat the following until the total number of applied iterations is more than $c\sqrt{N}$.
   (a) Initialize $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}} |i\rangle|y\rangle$. Consider an item i marked if and only if $T[i] < T[y]$.
   (b) Apply the quantum exponential searching algorithm of [2] to the first register, with a time-out of $2m_0$ iterations.
   (c) Observe the first register: let y' be the outcome. If $T[y'] < T[y]$, set threshold index y to y'.
3. Return y

For the analysis of the probability of success, let us assume that the algorithm runs long enough to find the minimum. We start by analyzing the expected time used to find the minimum.

Lemma 2 *At a particular execution of the algorithm – searching the minimum among M items – some indexes will be chosen as a threshold index and some not. Let $p(M,t)$ be the probability that the index holding the item of rank $t \in [1, M]$ is chosen. Then $p(M,t) = 1/t$.*

Proof by induction on M for a fixed t: The basis case $p(t,t) = 1/t$ is obvious. Assume that for all M between t and some k the equation is true. Then for $M = k+1$ at the first round, y is chosen uniformly among all M indexes, and can hold an item of rank either t, greater than t, or less than t. Only the former two cases contribute to the summation:

$$\begin{aligned} p(k+1,t) &= \frac{1}{k+1} + \frac{1}{k+1}\sum_{r>t}^{k+1} p(r-1,t) \\ &= \frac{1}{k+1} + \frac{k+1-t}{k+1}\cdot\frac{1}{t} = \frac{1}{t}. \end{aligned}$$

∎

*Supported in part by the ISI Foundation.

†LRI, bât 490, F–91405 Orsay, France. durr@lri.fr, http://www.lri.fr/~durr.

‡Supported by the ALCOM-IT Research Programme of the EU and the ISI Foundation.

§Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK–5230 Odense M, Denmark. u2pi@imada.ou.dk, http://www.imada.ou.dk/~u2pi.

The expected number of iterations used by the exponential searching algorithm of [2] to find the index of a marked item (among N items where $t-1$ items are marked) is at most $6\sqrt{N/(t-1)}$ (see [2]). Thus, this is also an upper bound for the expected number of iterations used by our searching algorithm to find the index of an item with rank less than the item of rank t. We can now deduce the expected running time to find the minimum by our algorithm:

Lemma 3 *The expected total number of iterations needed before y satisfies that $T[y]$ is minimum is at most $m_0 = 15\sqrt{N}$.*

Proof: The expected total number of iterations before $T[y]$ holds the minimum value is at most

$$\begin{aligned}\sum_{t=2}^{N}\frac{1}{t}6\sqrt{\frac{N}{t-1}} &= 6\sqrt{N}\sum_{t=1}^{N-1}\frac{1}{t+1}\frac{1}{\sqrt{t}} \\ &\leq 6\sqrt{N}\Big(\frac{1}{2}+\sum_{t=2}^{N-1}t^{-3/2}\Big) \\ &\leq 15\sqrt{N}.\end{aligned}$$

∎

Finally, we prove theorem 1.

Proof of theorem 1: By lemma 3, after at most $2m_0$ iterations, $T[y]$ holds the minimum value with probability at least $1/2$. Assume for a moment that we run the algorithm c times, each time for exactly $2m_0$ iterations. Let $y_1, \ldots, y_c$ be the outcomes. Let y be y_i such that $T[y_i]$ is minimum over all c outcomes. With probability at least $1-1/2^c$, $T[y]$ holds the minimum value.

The probability of success is even better if we run the algorithm only once for exactly $c2m_0$ iterations, because then we use the information provided by the previous iterations. ∎

If the values in T are not distinct, we use the same algorithm as for the simplified case. The analysis of the general case is unchanged, except that in lemma 2, the equation $p(N,t) = 1/t$ now becomes an inequality, $p(N,t) \leq 1/t$. Hence, the lower bound for the success probability given in theorem 1 for the simplified case is also a lower bound for the general case.

Acknowledgments

We wish to thank Stephane Boucheron for raising this problem to us and Richard Cleve for helpful discussions.

References

[1] Bennett, Charles H., Ethan Bernstein, Gilles Brassard and Umesh Vazirani, "Strengths and weaknesses of quantum computing", manuscript (1995).

[2] Boyer, Michel, Gilles Brassard, Peter Høyer and Alain Tapp, "Tight bounds on quantum searching", manuscript quant–ph/9605034 (1996).

[3] Grover, Lov K., "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (1996), 212–219.