On The Power of Exact Quantum Polynomial Time

Article · January 1997

Source: arXiv

On The Power of Exact Quantum Polynomial Time

Gilles Brassard [*] [†]
Université de Montréal

Peter Høyer [‡] [§]
Odense University

3 December 1996

Abstract

We investigate the power of quantum computers when they are required to return an answer that is guaranteed correct after a time that is upper-bounded by a polynomial in the worst case. In an oracle setting, it is shown that such machines can solve problems that would take exponential time on any classical bounded-error probabilistic computer.

1 Introduction

According to the modern version the Church–Turing thesis, anything that can be computed in polynomial time on a physically realisable device can be computed in polynomial time on a probabilistic Turing machine with bounded error probability. This belief has been seriously challenged by the theory of quantum computing. In particular, it was shown by Peter Shor that quantum computers can factor large numbers in polynomial time [11], which is conjectured to be impossible for classical devices. However, Shor's algorithm is polynomial-time in the expected sense: there is no upper bound on how long it will run on any given instance if we keep being unlucky. In this paper, we address the question of *Exact* Quantum Polynomial Time, which concerns the problems that quantum computers

[*] Supported in part by Canada's NSERC and Québec's FCAR.

[†] Département IRO, Université de Montréal, C.P. 6128, succursale centre-ville, Montréal (Québec), Canada H3C 3J7. email: brassard@iro.umontreal.ca.

[‡] Supported in part by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT). Research carried out while this author was at the Université de Montréal.

[§] Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK–5230 Odense M, Denmark. email: u2pi@imada.ou.dk.

can solve in guaranteed worst-case polynomial time with zero error probability. Note that this strong requirement would make randomness useless for classical machines: anything you can compute on a classical probabilistic computer with zero error probability in guaranteed worst-case polynomial time can be done in polynomial-time by a *deterministic* computer—simply run the probabilistic algorithm with an arbitrarily fixed sequence of coin "tosses".

The study of Exact Quantum Polynomial Time is not new. The very first algorithm ever designed to demonstrate an advantage of quantum computers over classical computers, due to Deutsch and Jozsa [9], was of this Exact nature. However, it solved a problem that could be solved just as efficiently on a classical probabilistic computer, provided an arbitrarily small (one-sided) error probability is tolerated. Here we demonstrate for the first time the existence of a relativized problem that can be solved in Exact Quantum Polynomial Time, yet it would require exponential time to obtain a correct answer with probability significantly better than 1/2 by any classical probabilistic (or deterministic) algorithm.

When it comes to decision problems, the well-known classical classes **P**, **ZPP** and **BPP** [10] give rise to their natural quantum counterparts **QP**, **ZQP** and **BQP**, respectively. A decision problem belongs to **QP** if it can be solved by a quantum algorithm whose answer is guaranteed correct and whose running time is guaranteed to be bounded by some fixed polynomial. It is in **ZQP** if it can be solved with zero error probability in quantum polynomial time: the answer is still guaranteed correct, but the running time is required to be polynomial merely in the expected sense (for each possible input). This corresponds to the classical notion of Las Vegas algorithms. Finally, the decision problem is in **BQP** if it returns the correct answer with probability better than 2/3 on all inputs, after a time that is bounded by a polynomial. (In this case, it makes no difference whether we consider expected or worst-case time.) This corresponds to the classical notion of Monte Carlo algorithms. Please note that the name "**EQP**" has been used by different authors, sometimes to mean **QP** [7] and sometimes to mean **ZQP** [4]. To avoid confusion, we shall refrain from using it at all.

The following results are known in quantum complexity theory: $\mathbf{P} \subseteq \mathbf{QP}$ [1, 2], $\mathbf{ZPP} \subseteq \mathbf{ZQP}$, $\mathbf{BPP} \subseteq \mathbf{BQP} \subseteq \mathbf{P}^{\#\mathbf{P}}$ [13, 4] and $\mathbf{BQP}^{\mathbf{BQP}} = \mathbf{BQP}$ [3]. It is believed that $\mathbf{ZQP} \not\subseteq \mathbf{BPP}$ because Shor's quantum factorization algorithm [11] allows to recognize

$$F \;=\; \{\langle x, y\rangle \mid x \text{ has a prime divisor smaller than } y\}$$

in **ZQP**, whereas if $F \in \mathbf{BPP}$ then factorization can be accomplished in polynomial expected time by a classical Las Vegas algorithm. Moreover, with appropriate oracles, it is known that $\mathbf{QP} \not\subseteq \mathbf{NP}$ [7] (and therefore $\mathbf{QP} \not\subseteq \mathbf{ZPP}$ [6]), $\mathbf{BQP} \not\subseteq \mathbf{BPP}$ [4, 12] and $\mathbf{NP} \cap \mathbf{co}\text{–}\mathbf{NP} \not\subseteq \mathbf{BQP}$ [3].

A major open question concerns the power of the weakest of all polynomial-time quantum classes, **QP**, compared to that of the strongest of all polynomial-time classical probabilistic classes, **BPP**. Could it be that **QP** $\subseteq$ **BPP**? Or perhaps rather **BPP** $\subseteq$ **QP**? Or are these two classes uncomparable? What about relativized versions of this question? Clearly any oracle under which **P** = **PSPACE** is so that **QP** = **BPP** as well, but what about oracles that separate **QP** from **BPP**? Even though this paper provides an oracle under which there is a problem that can be solved in Exact Quantum Polynomial Time but requires exponential time to be solved with probability better than 2/3 by any classical probabilistic algorithm, we do not solve the **QP** versus **BPP** question because our problem is not a decision problem. Unfortunately, there is no obvious way to turn our problem into a decision problem, in the way that the Deutsch-Jozsa algorithm (which did not concern a decision problem either) was turned into a relativized decision problem to separate **QP** from **NP** [7]. Thus we leave the separation of **QP** from **BPP** as an open problem.

We assume in this extended abstract that the reader is familiar with the basic notions of quantum computing [8, 5].

2 The Problem and its Quantum Solution

We consider a computational problem inspired from Simon's problem [12]. Let $n \geq 2$ be any given integer. Let $\oplus : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ denote the bitwise exclusive-or. Define a dot product $(\cdot) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ by $a \cdot b = (\sum_{i=1}^{n} a_i b_i) \bmod 2$ where $a = a_n \ldots a_1$ and $b = b_n \ldots b_1$.

Given: An integer $n \geq 2$ and a function $f : \{0,1\}^n \to \{0,1\}^{n-1}$.

Promise: There exists a nonzero element $s \in \{0,1\}^n$ such that for all $x, y \in \{0,1\}^n$, $f(x) = f(y)$ if and only if $x = y$ or $x = y \oplus s$.

Problem: Find a nonzero element $z \in \{0,1\}^n$ such that $s \cdot z = 0$.

Simon's original problem [12] is equivalent to the problem of determining the unknown string s. Our problem is reducible to that one, since if we know s we can easily find a nonzero element $z \in \{0,1\}^n$ with $s \cdot z = 0$. However, Simon's quantum algorithm cannot be used to solve our problem because it finds s in a time that is polynomial merely in the expected sense. There is a nice group-theoretic interpretation for our problem, and since that interpretation also helps simplify the notation, we shall use it. Hence, we reformulate the problem as follows.

Let $\mathbb{Z}_2 = \{0, 1\}$ denote the field of two elements. For any given integer $n \geq 2$, let G denote the group $\langle \mathbb{Z}_2^n, \oplus \rangle$. For any subgroup $K \leqslant G$, let $K^\perp = \{g \in G \,|\, g \cdot k = 0 \text{ for all } k \in K\}$ denote the orthogonal subgroup.

Given: An integer $n \geq 2$ and a function $f : G = \mathbb{Z}_2^n \to \{0, 1\}^{n-1}$.

Promise: There exists a subgroup $H = \{0, s\}$ of order 2 such that f is constant and distinct on each coset of H.

Problem: Find a nonzero member z of the orthogonal subgroup $H^\perp$.

For each value in the image of f, there are two preimages. More interesting and crucial for our algorithm we also have that, for each value $y \in \{0, 1\}^{n-2}$, there are exactly four values in G for which f evaluates to either $y0$ or $y1$. These four values form two distinct cosets of H.

Before giving the quantum algorithm for solving the problem, we state the notation used in the following. For any subset $A \subseteq G$, let $|A\rangle$ denote the equally-weighted superposition $\frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle$. In particular, if Hg is a coset of H, then $|Hg\rangle$ denotes the superposition $\frac{1}{\sqrt{2}}(|g\rangle + |g \oplus s\rangle)$.

Let $\mathbf{W}_2$ denote the one-bit Walsh-Hadamard transform, $\mathbf{W}_2 = \frac{1}{\sqrt{2}} \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix} \right)$, and let $\mathbf{W}_2^n$ denote the Walsh-Hadamard transform applied on each bit of a system of n bits. The result of applying $\mathbf{W}_2^n$ to an n–bit register $|w\rangle$ is the superposition $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{w \cdot x} |x\rangle$. Finally, let $\mathbf{Z} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix} \right)$ denote the conditional sign-shift transform. Our quantum algorithm uses no other transforms than $\mathbf{W}_2^n$, $\mathbf{Z}$, and those needed to evaluate the function f. The evaluation of function f is made reversible by mapping $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$. Note that a second application of this process will reset the second register since $|x\rangle|y \oplus f(x) \oplus f(x)\rangle = |x\rangle|y\rangle$. The complete quantum algorithm is as follows.

Orthogonal subgroup member algorithm

1. Initialize the system to be in the zero-state $|0\rangle|0\rangle$ where the first register is an n–bit register and the second is an $(n-1)$–bit register. Initially, apply the transform $\mathbf{W}_2^n$ to the first register, producing an equally weighted superposition of all elements in the group G, $\frac{1}{\sqrt{2^n}} \sum_{g \in G} |g\rangle|0\rangle$.

2. Compute f in quantum parallelism and store the result in the second register, producing $\frac{1}{\sqrt{2^n}} \sum_{g \in G} |g\rangle|f(g)\rangle$.

3. Measure all bits of the second register but the least significant, yielding

$$\tfrac{1}{\sqrt{2}}\Big(|Hg_1\rangle|f(g_1)\rangle + |Hg_2\rangle|f(g_2)\rangle\Big)$$

for some $g_1, g_2 \in G$ with $g_1 \oplus g_2 \notin H$.

4. Apply the conditional sign-shift transform $\mathbf{Z}$ on the least significant bit in the second register, producing (up to an overall sign-shift) $\tfrac{1}{\sqrt{2}}\Big(|Hg_1\rangle|f(g_1)\rangle - |Hg_2\rangle|f(g_2)\rangle\Big)$.

5. Compute f again in order to reset the second register, producing $\tfrac{1}{\sqrt{2}}\Big(|Hg_1\rangle - |Hg_2\rangle\Big)|0\rangle$.

6. Apply $\mathbf{W}_2^n$ to the first register again.

7. Measure the first register. Let $z^\star$ be the outcome.

We claim that this algorithm returns a nonzero member of the orthogonal subgroup, that is, that $z^\star$ is a nonzero element in $H^\perp$. Before proving this claim, we provide an example of the algorithm.

Example Let $n = 4$ and $G = \mathbb{Z}_2^4$. Let the unknown string s be 0101 and the unknown subgroup $H = \{0, s\}$. The function f is constant on each coset of H, so we need only specify it on a transversal T for H, say,

T	0000	0001	0010	0011	1000	1001	1010	1011
f	000	010	100	110	101	001	011	111

After the second step of the quantum algorithm, the system is in a superposition of all elements in the group, $\frac{1}{4}\sum_{g\in G}|g\rangle|f(g)\rangle$. Suppose we measure the string 01 in the third step. This projects the superposition to

$$\frac{1}{2}\bigg(\Big(|0001\rangle + |0100\rangle\Big)|010\rangle + \Big(|1010\rangle + |1111\rangle\Big)|011\rangle\bigg).$$

Applying the conditional sign-shift transform and uncomputing f produces

$$\frac{1}{2}\Big(|0001\rangle + |0100\rangle - |1010\rangle - |1111\rangle\Big)|000\rangle,$$

and applying the final Walsh-Hadamard transform gives the superposition

$$\frac{1}{2}\Big(|0010\rangle + |1000\rangle - |0101\rangle - |1111\rangle\Big)|000\rangle.$$

It can easily be verified that each of the four basis-states in this superposition holds a nonzero member of the orthogonal subgroup $H^\perp$ in the first register. □

Theorem 1 *Let $n \geq 2$ and $G = \mathbb{Z}_2^n$. Let $H \leqslant G$ be an unknown subgroup of order 2. Let $f : G \to \{0,1\}^{n-1}$ be any function constant and distinct on each coset of H. Then the quantum algorithm given above finds a nonzero member of $H^\perp$ in time polynomial in n and in the time to compute f.*

The theorem follows from the simple lemma below, by observing that only nonzero members of the orthogonal subgroup can have nonzero amplitude after completing step 6.

Lemma 2 *Let $G = \mathbb{Z}_2^n$ and $H = \{0, s\} \leqslant G$. Let $g_1, g_2 \in G$ be any two elements in G such that $g_1 \oplus g_2 \notin H$. Let*

$$|\psi\rangle = \mathbf{W}_2^n \left(\frac{1}{\sqrt{2}} \big(|Hg_1\rangle - |Hg_2\rangle \big) \right).$$

Then, for all $x \in G$,

$$\langle x|\psi\rangle = \begin{cases} \pm \frac{1}{\sqrt{2^{n-2}}} & \text{if } x \cdot s = 0 \text{ and } x \cdot (g_1 \oplus g_2) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Proof The amplitude of state $|x\rangle$ in superposition $|\psi\rangle$ is given by

$$\frac{1}{2\sqrt{2^n}} \left((-1)^{g_1 \cdot x} + (-1)^{(g_1 \oplus s) \cdot x} - (-1)^{g_2 \cdot x} - (-1)^{(g_2 \oplus s) \cdot x} \right).$$

This can be factorized as

$$\frac{1}{2\sqrt{2^n}} (-1)^{g_1 \cdot x} \left(1 + (-1)^{s \cdot x} \right) \left(1 - (-1)^{(g_1 \oplus g_2) \cdot x} \right),$$

and the lemma follows. □

3 Classical Lower Bound

In this section, we prove that any classical algorithm that would try to solve the above problem in subexponential time would have probability exponentially close to 1/2 to give a correct answer, which is essentially no better than guessing an answer at random. This is captured in the following theorem and its immediate corollary.

Theorem 3 *Consider an integer $n \geq 2$ and pick a function $f : \{0,1\}^n \to \{0,1\}^{n-1}$ at random according to the uniform distribution among all functions that satisfy the promise that there exists an $s \in \{0,1\}^n$ such that $f(x) = f(y)$ if and only if $x \oplus y = s$ for all distinct*

x and y in $\{0,1\}^n$. Consider an arbitrary classical algorithm that has access to f as an oracle. Assume the algorithm makes no more than $2^{n/3}$ calls on its oracle. Then there exists an event $\mathcal{E}$ such that (1) Prob$[\mathcal{E}] < 2^{-n/3}$ *and, (2) If $\mathcal{E}$ does* not *occur then the probability that the algorithm returns a nonzero $z \in \{0,1\}^n$ such that $s \cdot z = 0$ is less than $\frac{1}{2} + 2^{-n/3}$.*

Proof This theorem follows directly from Lemmas 6 and 7, which are stated and proven below. □

Corollary 4 *The probability that the algorithm mentioned in Theorem 3 will return a correct answer after making no more than $2^{n/3}$ calls on its oracle is less than $\frac{1}{2} + 1/2^{(n/3)-1}$.*

To establish these results, assume that the algorithm has queried its oracle on inputs $x_1, x_2, \ldots, x_k$ for $x_i \in \{0,1\}^n$, $1 \leq i \leq k \leq 2^{n/3}$. Without loss of generality, assume that all the queries are distinct. Let $y_1, y_2, \ldots, y_k$ be the answers obtained from the oracle, i.e. $y_i = f(x_i)$ for each i. Define the event $\mathcal{E}$ as *occurring* if there exist i and j, $1 \leq i < j \leq k$, such that $y_i = y_j$. Clearly, the algorithm has discovered the secret s when $\mathcal{E}$ occurs since in that case $s = x_i \oplus x_j$. This allows the algorithm to produce a correct solution with certainty. We have to prove that $\mathcal{E}$ is very unlikely and that, unless $\mathcal{E}$ occurs, the algorithm has so little information that it cannot return an answer that is significantly more probable to be correct than a random n–bit string.

Let $X = \{x_1, x_2, \ldots, x_k\}$ be the set of queries to the oracle and let $Y = \{y_1, y_2, \ldots, y_k\}$ be the corresponding answers. Let $W = \{x_i \oplus x_j \,|\, 1 \leq i < j \leq k\}$ and let $m < k^2$ be the cardinality of W. Note that $\mathcal{E}$ occurs if and only if $s \in W$ since $y_i = y_j$ if and only if $x_i \oplus x_j = s$. If $\mathcal{E}$ does not occur, we say that any nonzero n–bit string $\hat{s} \notin W$ is *compatible* with the available data because it is not ruled out as possible value for the actual unknown s. Similarly, given any compatible $\hat{s}$, we say that a function $\hat{f} : \{0,1\}^n \to \{0,1\}^{n-1}$ is *compatible* with the available data (and with $s = \hat{s}$) if $\hat{f}(x_i) = y_i$ for all i, and if $\hat{f}(x) = \hat{f}(y)$ if and only if $x \oplus y = \hat{s}$ for all distinct x and y in $\{0,1\}^n$. The following lemma says that all compatible values for s are equally likely to be correct given the available data, and therefore the only information available about s is that it is one of the compatible values.

Lemma 5 *Assume $\mathcal{E}$ has not occurred. There are exactly $(2^n - m - 1)((2^{n-1} - k)!)$ functions that are compatible with the available data. For each compatible string $\hat{s}$, exactly $(2^{n-1} - k)!$ of those functions are also compatible with $s = \hat{s}$.*

Proof Consider an arbitrary compatible $\hat{s}$. Define $X' = \{x \oplus \hat{s} \,|\, x \in X\}$. It follows from the compatibility of $\hat{s}$ that $X \cap X' = \emptyset$. Let $Z = \{0,1\}^n \setminus (X \cup X')$, where "\" denotes set difference. Note that $x \in Z$ if and only if $x \oplus \hat{s} \in Z$. Partition Z in an arbitrary way

into $Z_1 \cup Z_2$ so that $x \in Z_1$ if and only if $x \oplus \hat{s} \in Z_2$. The cardinalities of Z_1 and Z_2 are $(2^n - 2k)/2 = 2^{n-1} - k$. Now let $Y' = \{0,1\}^{n-1} \setminus Y$, also a set of cardinality $2^{n-1} - k$. To each bijection $h : Z_1 \to Y'$ there corresponds a function $\hat{f}$ compatible with the available data and $s = \hat{s}$ defined by

$$\hat{f}(x) \;=\; \begin{cases} y_i & \text{if } x = x_i \text{ for some } 1 \le i \le k \\ y_i & \text{if } x = x_i \oplus \hat{s} \text{ for some } 1 \le i \le k \\ h(x) & \text{if } x \in Z_1 \\ h(x \oplus \hat{s}) & \text{if } x \in Z_2. \end{cases}$$

The conclusion follows from the facts that there are $(2^{n-1} - k)!$ such bijections, each possible function compatible with the available data and $s = \hat{s}$ is counted exactly once by this process, and there are $2^n - m - 1$ compatible choices for $\hat{s}$, each yielding a disjoint set of functions compatible with the available data. □

Lemma 6 *Event $\mathcal{E}$ has probability of occurrence smaller than $2^{-n/3}$ provided the oracle is probed $k \le 2^{n/3}$ times.*

Proof Since all nonzero values for s are equally likely *a priori*, and since event $\mathcal{E}$ occurs if and only if $s \in W$, it follows that

$$\text{Prob}[\mathcal{E}] = m/(2^n - 1) < k^2/2^n \le 2^{-n/3},$$

where m is the cardinality of W. □

Lemma 7 *If event $\mathcal{E}$ does* not *occur then the probability that the algorithm returns a nonzero $z \in \{0,1\}^n$ such that $s \cdot z = 0$ is less than $\frac{1}{2} + 2^{-n/3}$, provided the oracle is probed $k \le 2^{n/3}$ times.*

Proof Assume that event $\mathcal{E}$ has not occurred after $k \le 2^{n/3}$ probes to the oracle. Consider an arbitrary nonzero $z \in \{0,1\}^n$. Let $A_z = \{u \in \{0,1\}^n \,|\, u \cdot z = 0\}$,

$$B_z = \{u \in A_z \,|\, u \neq 0^n \text{ and } u \notin W\}$$

and let b_z be the cardinality of B_z. It is well-known that A_z contains 2^{n-1} elements, and therefore $b_z \le 2^{n-1} - 1$. We know from Lemma 5 that the only knowledge about s that is available to the algorithm is that it is nonzero and not in W. Therefore, z is a correct answer if and only if $s \in B_z$, and the optimal strategy for the algorithm is to return some z

that maximizes b_z. Given that there are $2^n - 1 - m$ possible values for s, the probability of success (conditional to event $\mathcal{E}$ having not occurred) is

$$\frac{b_z}{2^n - 1 - m} \leq \frac{2^{n-1} - 1}{2^n - 1 - m} < \frac{2^{n-1}}{2^n - k^2} \leq \frac{2^{n-1}}{2^n - 2^{2n/3}} = \frac{1/2}{1 - 2^{-n/3}} \leq \frac{1}{2} + 2^{-n/3}$$

provided $n \geq 3$. The Lemma holds also when $n = 2$ since in this case at most one question is allowed ($2^{2/3} < 2$), which gives a success probability *smaller* than $1/2$ for all possible algorithms! □

4 Concluding Remarks and Open Problems

In the quantum algorithm, we performed a partial measurement at step 3. This step is not necessary, as we still will obtain a nonzero member of the orthogonal subgroup if we only perform steps 1, 2 and 4–7. We have, however, included it here to emphasize a group-theoretic interpretation of the algorithm. The algorithm (and the notion of orthogonal subgroups) can be generalized to arbitrary finite Abelian groups of smooth order. The requirement of smoothness is sufficient to be able to perform the quantum Fourier transform (step 1) and the conditional phase-changes (step 4) exactly in polynomial time.

Theorem 8 *Let G be any Abelian group of smooth order m. Let $H \leqslant G$ be an unknown subgroup of known index r, $r > 1$. Let $f : G \to \{0, \ldots, r-1\}$ be any function constant and distinct on each coset of H. Then there exists a quantum algorithm that finds a nonzero member of $H^\perp$ in time polynomial in $\log(m)$ and in the time to compute f.*

An interesting open question related to ours and Simon's algorithms is whether s can be found in Exact Quantum Polynomial Time. From a complexity-theoretic point of view, an oracle separation of **QP** and **BPP** is still an open question since our problem is not a decision problem.

References

[1] P. Benioff, "Quantum mechanical Hamiltonian models of Turing machines", *Journal of Statistical Physics*, Vol. 29, no. 3, 1982, pp. 515 – 546.

[2] C. H. Bennett, "Logical reversibility of computation", *IBM Journal of Research and Development*, Vol. 17, 1973, pp. 525 – 532.

[3] C. H. Bennett, E. Bernstein, G. Brassard and U. Vazirani, "Strengths and weaknesses of quantum computing", *SIAM Journal on Computing*, to appear.

[4] E. Bernstein and U. Vazirani, "Quantum complexity theory", *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 1993, pp. 11 – 20. Final paper to appear in *SIAM Journal on Computing*.

[5] A. Berthiaume, "Quantum computation", in *Complexity Theory Retrospective II*, L. Hemaspaandra and A. Selman (editors), Springer-Verlag, to appear.

[6] A. Berthiaume and G. Brassard, "The quantum challenge to structural complexity theory", *Proceedings of 7th Annual IEEE Structure in Complexity Theory Conference*, 1992, pp. 132 – 137.

[7] A. Berthiaume and G. Brassard, "Oracle quantum computing", *Journal of Modern Optics*, Vol. 41, 1994, pp. 2521 – 2535.

[8] G. Brassard, "A quantum jump in computer science", in *Computer Science Today*, Jan van Leeuwen (editor), Lecture Notes in Computer Science, Vol. 1000, Springer-Verlag, 1995, pp. 1 – 14.

[9] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation", *Proceedings of the Royal Society, London*, Vol. A439, 1992, pp. 553 – 558.

[10] J. Gill, "Computational complexity of probabilistic Turing machines", *SIAM Journal on Computing*, Vol. 6, 1977, pp. 675 – 695.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring", *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*, 1994, pp. 124 – 134. Final version to appear in *SIAM Journal on Computing* under title "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer".

[12] D. R. Simon, "On the power of quantum computation", *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*, 1994, pp. 116 – 123.

[13] L. Valiant, personal communication through [4], 1992.