Proseminar Randomisierte Methoden

Quantum Computing & Algorithms

Loginov Oleg

May 27, 2004

Abstract

This is an introductory review of quantum calculations. Quantum Logic and some of aspects of quantum computing are given here. Two main quantum algorithms are also represented. It is intended for non-specialists which have basic knowledge on undergraduate Linear Algebra. It's common review, and, for further study you should use another, more specialized, literature.

Contents

1 Introduction

I have put all efforts to write as clear as possible for non-specialists, but i don't think that this paper is easy for comprehension. Section 2 contain basics for quantum computing, like quantum logic and controlling operations. All it are need for further understanding of using it in algorithms. I assume familiarity with undergraduate Linear Algebra, which is the main mathematical basis of Quantum Mechanics. Section 3 show the first quantum algorithm, Shor's Algorithm, it was developed for finding the prime factors of a composite number N. Section 4 is dedicated to algorithm for search task, Grover's Algorithm. I don't give practical scheme or realization of algorithms, because it is not the theme of this paper and I can give only references on it.

2 Foundations

In classical computers, a bit can assume only values 0 or 1. In quantum computers, the values 0 and 1 are replaced by vectors $|0\rangle$ and $|1\rangle$. This notation for vectors is called Dirac Notation and is standard in Quantum Mechanics. The name bit is replaced by *qubit*, short of *quantum bit*. The difference between bits and qubits is fact that a qubit $|\psi\rangle$ can also be in a linear composition of vector $|0\rangle$ and $|1\rangle$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where α and β are complex numbers, $|\psi\rangle$ is said to be a *superposition* of the vectors $|0\rangle$ and $|1\rangle$ with *amplitudes* α and β. Thus, $|\psi\rangle$ is a vector in two-dimensional complex vector space, where $\{|0\rangle, |1\rangle\}$ forms an orthonormal basis, called the *computational basis*. The state $|0\rangle$ is not the zero vector, but simply the first vector of the basis. The matrix representations of the vectors $|0\rangle$ and $|1\rangle$ are given by

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

In Quantum Mechanics, vectors are systematically called *states*.

The physical interpretation of $|\psi\rangle$ is that it coexists in two states: $|0\rangle$ and $|1\rangle$. It is similar to a coin that is partially heads up and partially tails up simultaneously. We cannot push further analogy simply because quantum phenomena do not have a classical analogue in general.

The state $|\psi\rangle$ can store a huge quantity of information in its coefficients α and β, but this information lives in quantum level, which is microscopic (usually quantum effects appear in atomic dimensions). To bring quantum information to classical level, one must measure a qubit. Quantum Mechanics tells us that the measurement process inevitably disturbs a qubit state, producing a non-deterministic collapse of $|\psi\rangle$ to either $|0\rangle$ or $|1\rangle$. One gets $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$. The non-deterministic collapse does not allow one to determine the values of α and β before the measurement. They are inaccessible via measurements unless one has many copies of the same state. Two successive measurements of the same qubit give the same output. If $|\alpha|^2$ and $|\beta|^2$ are probabilities and there are only two possible outputs, then

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2}$$

A measurement is not the only way that one can interact with a qubit. If one does not obtain any information about the state of the qubit, the interaction changes the values of α and β keeping the constraint (2). The most general transformation of this kind is a linear transformation U that takes unit vectors into unit vectors. Such transformation is called unitary and can be defined by

$$U^\dagger U = UU^\dagger = I,$$

where $U^\dagger = (U^*)^T$ ($*$ indicates complex conjugation and T indicates the transpose operation) and I is the 2×2 identity matrix.

So far we are dealing with one-qubit quantum computers. To consider the multiple qubit case, it is necessary to introduce the concepts of *tensor product*. Suppose V and W are complex vector spaces of dimensions m and n, respectively. The tensor product $V \otimes W$ is an mn-dimensional vector space (generally, there are several conditions for this space, you can know more about it by yourself). I use also the notations $|v\rangle|w\rangle$, $|v, w\rangle$ or $|vw\rangle$ for the tensor product

$$|v\rangle \otimes |w\rangle.$$

Note that the tensor product is non-commutative, so the notation must preserve the ordering.

Given two linear operators A and B defined on the vector spaces V and W, respectively, we can define the linear operator $A \otimes B$ on $V \otimes W$ as

$$(A \otimes B)(|v\rangle \otimes |w\rangle) = A|v\rangle \otimes B|w\rangle \tag{3}$$

where $|v\rangle \in V$ and $|w\rangle \in W$. The matrix representation of $A \otimes B$ is given by

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1m}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mm}B \end{bmatrix} \tag{4}$$

where A is $m \times m$ matrix and B is $n \times n$ matrix (using the same notation for the operator and its matrix representation). So, the matrix $A \otimes B$ has dimension $mn \times mn$. The formula (4) can also be used for non-square matrices, such as the tensor product of two vectors. For example, if we have a 2-qubit quantum computer and the first qubit is in the state $|0\rangle$ and the second is in state $|0\rangle$, then the quantum computer is in the state $|0\rangle \otimes |1\rangle$, given by

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

The resulting vector is in a 4-dimensional vector space. The general state $|\psi\rangle$ of a 2-qubit quantum computer is a superposition of the states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$,

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \tag{6}$$

with the constraint

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Regarding the zeroes and ones as constituting the binary expansion of an integer, we can replace the representations of states $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, by the shorter forms $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$ in decimal forms, which is handy in some formulas.

The state of an n-qubit quantum computer is a vector in a 2^n-dimensional complex vector space. When the number of qubits increase linearly, the dimension of the associate vector space increase exponentially. As before, a measurement of a generic state $|\psi\rangle$ yields the result $|i_0\rangle$ with probability $|\alpha_{i_0}|^2$, where $0 \leq i_0 \leq 2^n$. Usually, the measurement is performed qubit by qubit yielding zeroes or ones that are read together to form i_0. I stress again a very important feature of measurement process. The state $|\psi\rangle$ as it is before measurement is inaccessible unless it is in computational basis. The measurement process inevitably disturbs $|\psi\rangle$ forcing it to collapse to one vector of the computational basis. This collapse is non-deterministic, with the probabilities given by squared norms of the corresponding amplitudes in $|\psi\rangle$.

The general 2-qubit state is not necessarily a product of two one-qubit states. Such non-product states of two or more qubits are called *entangled* states, for example, $(|00\rangle + |11\rangle)/\sqrt{2}$.

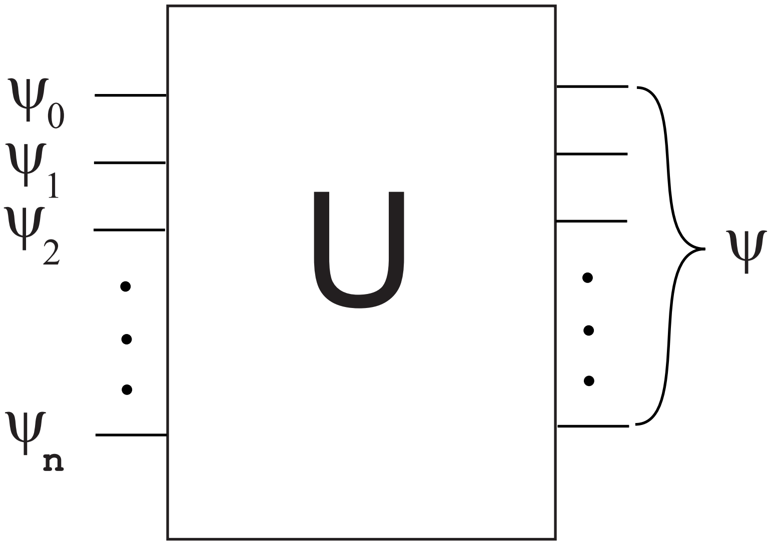

Figure 1: Quantum computer.

The entangled states play an essential role in quantum computers. On the other hand, a naive use of entanglement does not guarantee any improvements.

After the above review, we are ready to outline the quantum computer. In Fig. 1, we are taking a non-entangled input, what is quite reasonable. In fact, $|\psi_i\rangle$ is either $|0\rangle$ or $|1\rangle$ generally. $|\psi\rangle$, on the right hand side of figure, is the result of the application of unitary operation U on the input. The last step is the measurement of the states of each qubit, which returns zeroes and ones that form the final result of the quantum calculation. Note that there is, in principle, an infinite number of possible operators U, which are unitary $2^n \times 2^n$ matrices.

Let's speak about Quantum Circuits (Gates), and start with one-qubit gates.

In the classical case there is only one possibility, which is the *NOT* gate. The *NOT* gate inverts the bit value: 0 goes 1 and vice-versa. The straightforward generalization to the quantum case is given in fig 3, where X is the unitary operator

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

So, if input $|\psi\rangle$ is $|0\rangle$, the output is $|1\rangle$ and vice-versa. But now we can have a situation with no classical counterpart. The state $|\psi\rangle$ can be in superposition of states $|0\rangle$ and $|1\rangle$. The general case is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and the corresponding output is $\alpha|1\rangle + \beta|0\rangle$.

The *Hadamard* gate is another important one-qubit gate, given by

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

It is easy to see that $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$; $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. If input is $|0\rangle$, the Hadamard gate creates a superposition of states with equal weights. This is a general feature, valid for two or more qubits. Thus, the tensor product of n Hadamard operators produces an equality weighted superposition of all computational basis states, when the input is the state $|0\rangle$.

Another important 2-qubit quantum gate is *CNOT* gate. It has two input qubits, the control and the target qubit, respectively. The target qubit is flipped only if the control qubit is set to 1. The action of the *CNOT* gate can also be represented by

$$|a, b\rangle \to |a, a \oplus b\rangle,$$

where $\oplus$ is addition modulo 2.

$CNOT$ and the one-qubit gates form a universal set of gates. This means that any other gate, operating on 2 or more qubits, can be written as compositions and direct products of $CNOT$ and one-qubit gates.

3 Shor's Algorithm

Let us describe Shor's algorithm for finding the prime factors of a composite number N. Think of a large number such as one with 300 digits in decimal notation, since such numbers are used in cryptography. Though N is large, the number of qubits necessary to store it is small. In general $\log_2 N$ is not an integer, so let us define

$$n = [\log_2 N].$$

Reduction of factorization of N to the problem of finding the order of an integer x less than N is as follows. If x and N have common factors, then $GCD(x, N)$ gives a factor of N, therefore it suffices to investigate the case when x is coprime to N. The order of x modulo N is defined as the least positive integer r such that

$$x^r \bmod N \equiv 1$$

If r is even, we can define y by

$$x^{r/2} \bmod N \equiv y$$

The above notation means that y is the remainder of $x^{r/2}$ divided by N and, by definition, $0 \leq y \leq N$. Note that y satisfies $y \bmod N \equiv 1$, or equivalently $(y-1)(y+1) \bmod N \equiv 0$, which means that N divides $(y-1)(y+1)$. If $1 < y < N-1$, the factors $y-1$ and $y+1$ satisfy $0 < y-1 < y+1 < N$, therefore N cannot divide $y-1$ nor $y+1$ separately. The only alternative is that both $y-1$ and $y+1$ have factors of N (that yield N by multiplication). So, $GCD(y-1, N)$ and $GCD(y+1, N)$ yield non trivial factors of N (GCD stands for the greatest common divisor). If N has remaining factors, they can be calculated applying the algorithm recursively.

Consider $N = 21$ as an example. The sequence of equivalences

$$\begin{array}{ll} 2^4 & \equiv 16 \bmod 21 \\ 2^5 & \equiv 11 \bmod 21 \\ 2^6 & \equiv 11 \times 2 \equiv 16 \bmod 21 \end{array}$$

show that the order of 2 mod 21 is $r = 6$. Therefore, $y \equiv 2^3 \equiv 8 \bmod 21$. $y-1$ yields the factor 7 and $y+1$ yields the factor 3 of 21.

In summary, if we pick up at random a positive integer x less than N and calculate $GCD(x, N)$, either we have a factor of N or we learn that x is coprime to N. In the latter case, if x satisfies the conditions (1) its order r is even, and (2) $0 < y-1 < y+1 < N$, then $GCD(y-1, N)$ and $GCD(y+1, N)$ yield factors of N. If one of the conditions is not true, we start over until finding a proper candidate x. The method would not be useful if these assumptions were too restrictive, but fortunately that is not the case. The method systematically fails if N is a power of some odd prime, but an alternative efficient classical algorithm for this case is known. If N is even, we can keep dividing by 2 until the result turns out to be odd. It remains to apply the method for odd composite integers that are not a power of some prime number. It is cumbersome to prove that the probability of finding x coprime to N satisfying the conditions (1) and (2) is high; in fact this probability is $1 - 1/2^{k-1}$, where k is the number of prime factors of N. In the worst case (N has 2 factors), the probability is greater than or equal to $1/2$.

At first sight, it seems that we have just described an efficient algorithm to find a factor of N. That is not true, since it is not known an efficient classical algorithm to calculate the order of an integer $x \ mod \ N$. On the other hand, there is (after Shor's work) an efficient quantum algorithm.

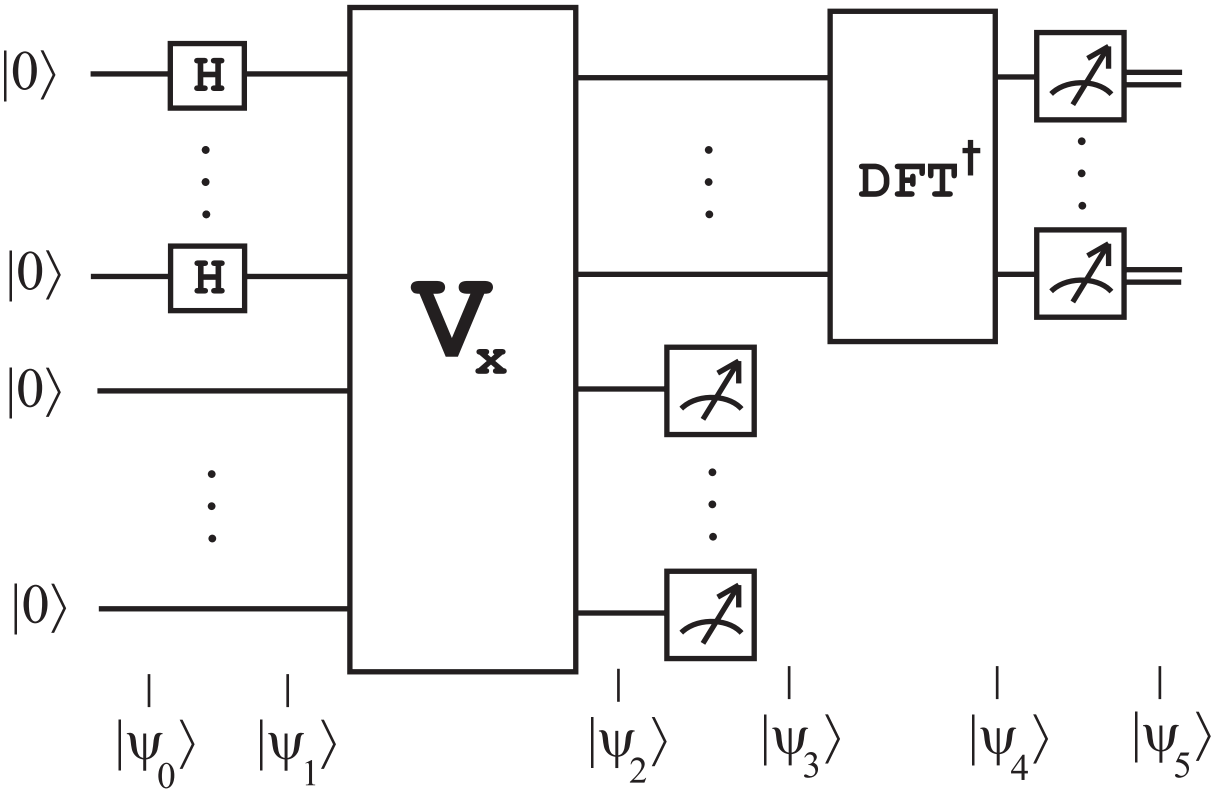

Figure 2: Scheme of Shor's algorithm.

Consider the circuit of Fig 2. It calculates the order r of the positive integer x less than N, coprime to N. V_x is the unitary linear operator

$$V_x(|j\rangle|k\rangle) = |j\rangle|k + x^j\rangle, \tag{7}$$

where $|j\rangle$ and $|k\rangle$ are the states of the first and second registers, respectively. The arithmetical operations are performed modulo N, so $0 \leq k + x^j \leq N$. DFT is the Discrete Fourier Transform operator.

The first register has t qubits, where t is generally chosen such that $N^2 \leq 2^t \leq 2N^2$. As an exception, if the order r is a power of 2, then it is enough to take $t = n$. In this section I consider this very special case and leave the general case for other discussion.

The states of the quantum computer are indicated by $|\psi_0\rangle$ to $|\psi_5\rangle$ in fig. 5. The initial state is

$$|\psi_0\rangle = |0..(t \ times)..0\rangle|0..(n \ times)..0\rangle.$$

The application of the Hadamard operator on each qubit of the first register yields

$$|\psi_1\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|0\rangle. \tag{8}$$

The first register is then in a superposition of all states of the computational basis with equal amplitudes given by $1/\sqrt{2^t}$. Now we call attention to what happens when we apply V_x to $|\psi_1\rangle$:

$$|\psi_2\rangle = V_x|\psi_1\rangle = \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1} V_x(|j\rangle|0\rangle) = \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|x^j\rangle. \tag{9}$$

The state $|\psi_2\rangle$ is a remarkable one. Because V_x is linear, it acts on all $|j\rangle|0\rangle$ for 2^t values of j, so this generates all powers of x simultaneously. This feature is called quantum parallelism. *Quantumly*, one can calculate all powers of x with just one application of V_x. At the quantum level, the values of j that yield x^j mod $N \equiv 1$ are "known". But this quantum information is not fully available at the classical level. A classical information of a quantum state is obtained by practical measurements and, at this point, it does not help if we measure the first register, since all states in the superposition (9) have equal amplitudes. The first part of the strategy to find r is to observe that the first register of the states $|0\rangle|1\rangle$,$|r\rangle|1\rangle$,$|2r\rangle|1\rangle$, ..., $|2^t - r\rangle|1\rangle$ is periodic. So the information we want is a period. In order to simplify the calculation, let us measure the second register. Before doing this, we will rewrite $|\psi_2\rangle$ collecting equal terms in the second register. Since x^j is a periodic function with period r, substitute $ar + b$ for j in Eq. 9, where $0 \leq a \leq (2t/r) - 1$ and $0 \leq b \leq r - 1$. Recall that we are supposing that $t = n$ and r is a power of 2, therefore r divides $2t$. Eq. 9 is converted to

$$|\psi_2\rangle = \frac{1}{\sqrt{2^t}}\sum_{b=0}^{r-1}\left(\sum_{a=0}^{\frac{2^t}{r}-1}|ar+b\rangle\right)|x^b\rangle. \tag{10}$$

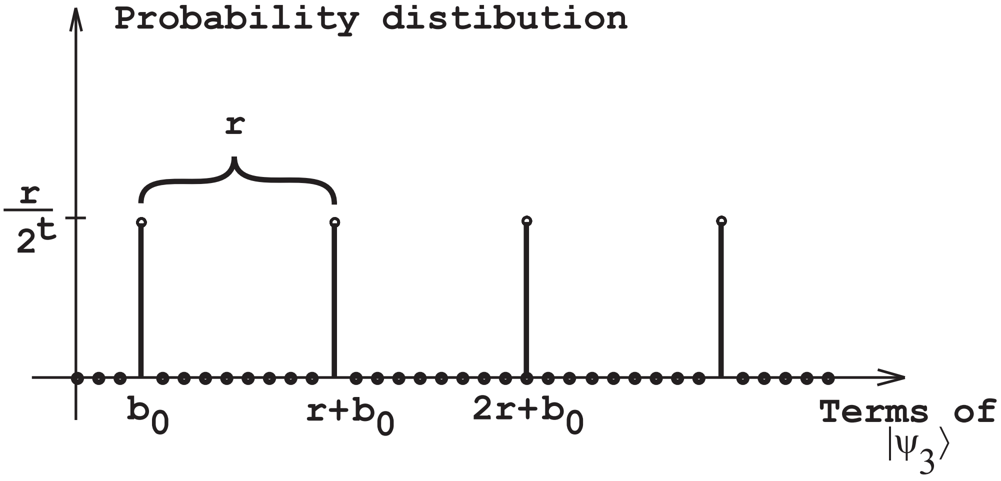

Figure 3: Scheme of Shor's algorithm.

In the second register, we have substituted x^b for x^{ar+b}, since $x^r \equiv 1$ *mod* N. Now the second register is measured. Any output x^0, x^1, ..., x^{r-1} can be obtained with equal probability. Suppose that the result is x^{b_0} . The state of the quantum computer is now

$$|\psi\rangle = \sqrt{\frac{r}{2^t}}\left(\sum_{a=0}^{\frac{2^t}{r}-1}|ar+b_0\rangle\right)|x_0^b\rangle. \tag{11}$$

Note that after the measurement, the constant is renormalized to $\sqrt{r/2^t}$, since there are $2^t/r$ terms in the sum (11). Fig 3 shows the probability of obtaining the states of the computational basis upon measuring the first register.

The probabilities form a periodic function with period r. Their values are zero except for the states $|b_0\rangle$, $|r+b_0\rangle$, $|2r+b_0\rangle$, ..., $|2^t-r+b_0\rangle$. How can one find out the period of a function efficiently? The answer is in the Fourier transform. The Fourier transform of a periodic function with period r is a new periodic function with period proportional to $1/r$. This makes a difference for finding r. The Fourier transform is the second and last part of the strategy. The whole method relies on an efficient quantum algorithm for calculating the Fourier transform, which is not available classically.

4 Grover's Algorithm

Let a system have $N = 2^n$ states which are labelled S_1,S_2,...,S_N. These 2^n states are represented as n bit strings. Let there be a unique state, say S_ν, that satisfies the condition $C(S_\nu) = 1$, whereas for all other states S, $C(S) = 0$ (assume that for any state S, the condition $C(S)$ can be evaluated in unit time). The problem is identify the state S_ν.

This could represent a database search problem where the function $C(S)$ is based on the contents of memory location corresponding to state S. Alternatively, if could represent a problem where the function $C(S)$ was being evaluated by the computer.

Steps (1) and (2) are a sequence of elementary unitary operations.Step (3) is the final measurement by an external system.

1. Initialize the system to superposition: $\left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots, \frac{1}{\sqrt{N}}\right)$, i.e. there is the same amplitude ti be in each of the N states. This superposition can be obtained in $O(log(N))$ steps.
2. Repeat the following unitary operations $O(\sqrt{N})$ times:
   (a) Let the system be in any state S: in the case $C(S) = 1$, rotate the phase by π radians; in the case $C(S) = 0$, leave the system unaltered.
   (b) Apply the diffusion transform D which is defined by matrix D as follows: $D_{ij} = \frac{2}{N}$ if $i \neq j$ and $D_{ij} = -1 + \frac{2}{N}$. This D-transform can be implemented as product of 3 elementary operators as discussed later.
3. Measure the resulting state, This will be the state S_ν (i.e. the desired state that satisfies the condition $C(S_\nu) = 1$) with probability of at least 0.5.

Note that step 2(a) is a phase rotation transformation. In an implementation it would involve a portion of the quantum system sensing the state and then deciding whether or not to rotate the phase. It would do it in a way so that no trace of the state of the system be left after this operation so as to ensure that paths leading to the same final state were indistinguishable and could interfere.

The loop in step 2 above, is the heart of algorithm. Each iteration of this loop increases the amplitude in the desired state by $O(1/\sqrt{N})$. As a result in $O(\sqrt{N})$ repetitions of the loop, the amplitude and hence the probability in the desired state reach $O(1)$. In order to see that the amplitude increases by $O(1/\sqrt{N})$ in each repetition, we first show that the diffusion transform, D, can be interpreted as an *inversion above average* operation. This operation, as described below more precisely, is also unitary operation, and is equivalent to the diffusion transform D as used in step 2(a).

Let α denote the average amplitude of over all states, i.e. if α_i be the amplitude in i^{th} state, then the average is $\frac{1}{N}\sum_{i=1}^{N}\alpha_i$. As a result of the operation D, the amplitude in each state increases (decreases) so that after this operation it is as much below (above) α. as it was above (below) α before the operation.

The diffusion transform, D, is defined as follows: $D_{ij} = \frac{2}{N}$, if $i \neq j$ and $D_{ii} = -1 + \frac{2}{N}$. Observe that D can be represented in the form $D \equiv -I + 2P$ where I is the identity matrix and P is a projection matrix with $P_{ij} = \frac{1}{N}$ for all i, j. The following two properties of P are easily verified: first, that $P^2 = P$, and second, that P acting on any vector $\overline{\nu}$ gives a vector, each of whose components is equal to the average of all components.

Using the fact that $P^2 = P$, it follows immediately from the representation $D = -I + 2P$ that $D^2 = I$ and hence D is unitary.

In order to see that D is is the *inversion about average*, consider what happens when D acts on arbitrary vector $\overline{\nu}$. Expressing D as $-I+2P$, it follows that: $D\overline{\nu} = (-I+2P)\overline{\nu} = -\overline{\nu}+2P\overline{\nu}$. By the discussion above, each component of vector $P\overline{\nu}$ is A where A is the average of all components of vector $\overline{\nu}$. Therefore the i^{th} component of the vector $D\overline{\nu}$ is given by $(-\nu_i + 2A)$ which can be written as $(A + (A - \nu_i))$ which is precisely the inversion about average.

Next consider the situation in figure 2, when this operation is applied to a vector with each of the components, except one, having an amplitude equal to $\frac{C}{\sqrt{N}}$ where C lies between $\frac{1}{2}$ and 1; the one component that is different has an amplitude of $-\sqrt{1-C^2}$.

The average A of all components is approximately equal to $\frac{C}{\sqrt{N}}$. Since each of the $(N-1)$ components is approximately equal to average, they do not change significantly as a result of the inversion about average. The one component that was negative, now becomes positive and it's magnitude increases by $\frac{2C}{\sqrt{N}}$.

In the loop of step 2, first the amplitude in the selected state is inverted (this is a phase rotation and hence a valid quantum mechanical operation). The the inversion about average operation is carried out. This increases the amplitude in the selected state in each iteration by $2C/\sqrt{N}$. Therefore as long as the magnitude of amplitude in the single state, i.e. $\sqrt{1-C^2}$, is less than $\frac{1}{\sqrt{2}}$, the increase in its magnitude is greater than $\frac{1}{\sqrt{2N}}$. It immediately follows that there exists a number M less than $\sqrt{N}$, such that in M repetitions of the loop in step 2, the magnitude of the amplitude in the desired state will exceed $\frac{1}{\sqrt{2}}$. Therefore if the state of a system is now measured, it will be in the state with probability greater than 0.5.

Here is no one practical realization of this algorithm. Anyway, you can find one in [3].

References

[1] C. Lavor†, L.R.U. Manssur‡, and R. Portugal‡. Shor's Algorithm for Factoring Large Integers. arXiv:quant-ph/0303175

[2] Lov K. Grover. Quantum Mechanics helps in searching for a needle in a haystack.

[3] Jennifer L. Dodd, Timothy C. Ralph and G. J. Milburn. Experimental requirements for Grover's algorithm in optical quantum computation. arXiv:quant-ph/0306081

[4] Sangwoong Park* ,Joonwoo Bae†, Younghun Kwon‡. Wavelet Quantum Search Algorithm with Partial Information. arXiv:quant-ph/0303025