Introduction to Quantum Computing

Paramjot Singh **Laksh Advani**

CSCI 5454 : Design and Analysis of Algorithms

Computer Science Department, University of Colorado Boulder

paramjot.singh@colorado.edu

laksh.advani@colorado.edu

Abstract

This paper introduces the concept of Quantum Computing along with the basics and phenomenon required to understand Quantum mechanics. As per the definition, Quantum Computing is a method of computing that uses quantum mechanical features like superposition and entanglement to achieve computation with 'Quantum Bits'. Quantum Algorithms are algorithms which run on a model of 'Quantum Computation'. After the introduction, we will investigate some of the key Quantum Algorithms that demonstrates the differences and benefits of Quantum Computers have over Classical Computing. This paper also introduces preliminary concepts like 'Superposition', Qubits and 'Hadamard Gates' to name a few. Next, we demonstrate Simon's algorithm to show the advantages quantum computers have over classical systems. Also, we use Grover's algorithm to exhibit how a quantum system can search an unsorted database in $\sqrt{N}$ time. We conclude by discussing about the caveats in Quantum computing and we will discuss the future of computing if Quantum computers becomes scalable, which is most likely in next 10 years or so.

Keywords — *Quantum Computing; Quantum Mechanics; Grover's Algorithm; Simon's Algorithm; Qubits; Superposition; Black Box; Quantum Gates; Entanglement; Quantum Search; Hadamard Gates;*

I. Introduction

Quantum computation starts with remarkable discovery that quantum systems are exponentially powerful. The major goal of Quantum computation is to harness this exponential power to solve interesting problems, which are otherwise difficult to solve with classical Turing systems. In this paper, we will introduce quantum mechanics and how quantum computation works. Next, we will understand how to harness this power using the examples of two famous quantum algorithms. Finally, we'll conclude by briefly discussing what are the limits and future of Quantum computers.

II. Background

This history of quantum mechanics starts with the double slit experiment performed by American physicists Davisson and Germer in 1927, which facilitated the discovery of the dual nature of light and other sub-atomic particles such as electrons, which is the principle unit in quantum computation. After that experiment, different physicists started studying quantum mechanics and the possible applications of the phenomenon. Till the end of 1970s many different scientists published papers discussing about possibilities of using this for computation.

In early 1980s, Yuri Manin first proposed an idea of quantum computer in his book. This started the new wave of publications focusing around the similar notes for quantum computer. Later in 1981, Richard Feynman at MIT talk proposed basic model of a quantum computer. Just after one year after that Paul Benioff introduced the first theoretical framework for quantum computer. Later for the decade many other inventions happened such as application of entanglement for secure communication, application of conjugate coding in distributing cryptographic keys and theorems like no-cloning theorem and different oracle problem.

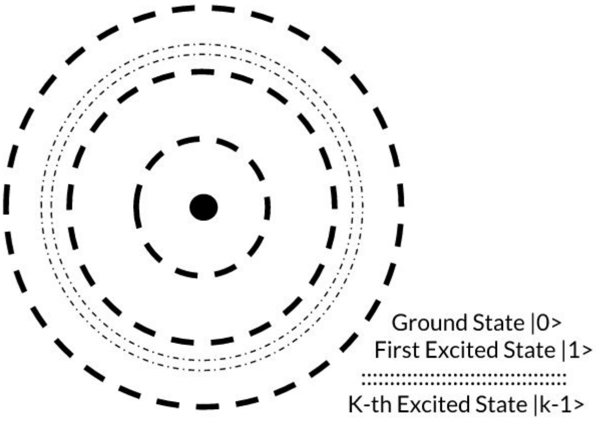
Next big step happened in the mid 90s, when Peter Shor developed an algorithm for prime factorization of large integers in polynomial time, which is currently an exponential problem. Hence, this algorithm can technically break most of the cryptosystems used in current security system (RSA algorithms etc). After this another monumental discovery occurred in just next two years, when Lov Kumar Grover invented quantum algorithm for database search with quadratic speedup. Main advantage of this algorithm was wide variety of problems which can be solved by this algorithm such as Circuit problem, SAT problem etc.

Later in 1998, the first 2-qubit quantum computer was created based on NMR architecture and later a 3-qubit. Same year execution was performed of Grover's algorithm on NMR computer. Next in 2000s, development of better quantum computers took place and many existing and new problems were solved in quantum space and as we're writing this report new inventions are happening in this field. Now all the major companies, IBM, Intel, Google, Microsoft, D-Wave, are working on developing and leading this field of computing.

III. Basics

A. Qubit

The Qubit or Quantum bit is the basic unit of quantum information. Let's start with quick recap about physics of electron in an atom and its energy levels.



Let's assume we have an atom, it's electron is in one of the several discrete state. For example, it can be in the ground state, or in first excited state, or second excited state, etc. For simplicity let's take case of a Hydrogen atom, so we have the electron in either the ground state or the first excited state (or just excited state). Now, we can encode the bit by saying that the ground state encodes for 0 and excited state encodes for 1.

But, the problem is the whole system and representation is not that easy. Since, an electron in an atom is not certainly in any specific state and hence, ends up being partly in ground and partly in the excited state. Hence, we can represent it by specifying that the electron is in ground state with α probability and electron is in the excited state with β probability. But in fact, what quantum mechanics tell us that the electron ends up not in one specific state but instead it ends up in superposition of ground and excited state with complex amplitudes α and β for ground (|0>) and excited state (|1>) respectively, where the sum of magnitude square is always unity.

B. Superposition

Superposition is one of the fundamental principles of quantum mechanics which states that any two or more quantum states can be superposed together and the resulting state will be another valid quantum state and converse holds true as well. Now, with respect to our ground and excited state example for H atom. We can represent those complex amplitude α and β and states 0 and 1, by using following quantum state Ψ, such that

$$|\Psi> = \alpha\ |0> + \beta\ |1>$$

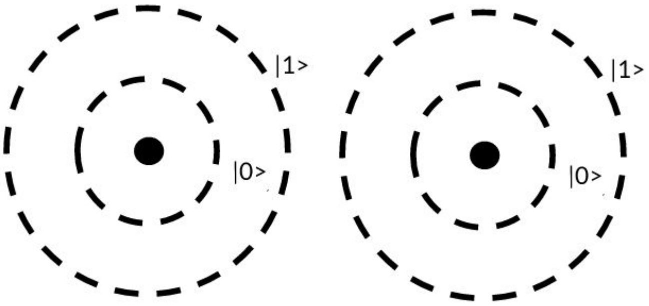
$$|\alpha|^2 + |\beta|^2 = 1$$

So this says about state of electron that electron is in superposition with complex normalized amplitude. But, as soon as we look at it or take measurement of its state, it goes to either to ground or excited state with probability $|\alpha|^2$ or probability $|\beta|^2$, respectively. Hence, we disturb the state |Ψ> when we look or try to measure the system. But, it is little confusing to understand that what could it mean for electron to be in some state with some complex amplitude, which doesn't make any sense but only good thing is that we have the mathematical notation of it and hence can work on developing an intuition for these peculiar state of affairs.

Also notice one more thing, since I mentioned that electron is in this superposition state as long as we don't try to measure its state. So, amount of information it takes to specify the state of electron when we are not disturbing the system is two complex numbers, which is infinite number of bits of information. So, it's a very complicated state that describes what state of electron is when system is not disturbed. But when we try to look at it, it drops down to single bit information of either ground or excited state.

C. Entanglement

Entanglement is another fundamental phenomenon from quantum mechanics that occurs when pairs of qubits are interacting in a way such that quantum state of each particle can't be described independently of the others, even when they are seperated after keeping them in a combined system.



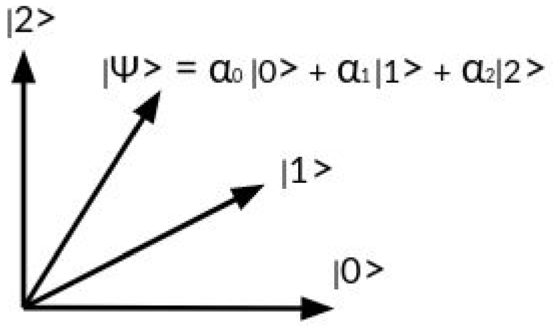
Let's take example with 2 qubit system of H atom and we can represent the entangled quantum state of the combined state as

$$|\Psi> = \alpha_{00}\,|00> + \alpha_{01}\,|01> + \alpha_{10}\,|10> + \alpha_{11}\,|11>$$

This is because of the property of quantum system that if we bring two quantum system close to each other and let them interact, then they can get it into a state such that neither or these systems can be described by itself. Hence, they can be described by describing all of it and this entangled state stays even after we take it apart now with large distance.

D. Geometric Representation of K-level System

So, geometrically the superposition we understand as the state of k-level quantum system is a unit vector in k-dimensional complex vector space which is also called Hilbert Space C^k

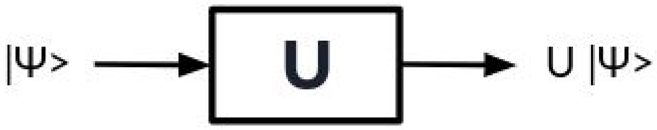


So, when we measure the |Ψ> system, then Ψ gets projected to one of the bases |0>, |1> or |2>. For example, if we are measuring for |0> base, then Ψ is projected with probability of $cos^2\theta$.

P[0] = $cos^2\theta$ and the new state |Ψ'> = |0>, since the measurement disturbed the system.
Similarly we can go for projection in other bases including the sign base |+> or |->, which are equal probable of both |0> and |1>.

IV. Quantum Gates

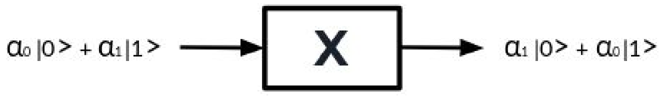
Like we have classical gates, such as NOT, AND, NOR, etc. We have similar gates for quantum systems. The classical gates can't be used with qubit system, since to perform classical gate action, we'll need to do measurement of information and hence in-turn will lose the superposition and will be merely traditional bits. So, in quantum gates we have input as some quantum state |Ψ> and output as u|Ψ>



Following are the basic quantum gates we use for most of the operations using qubits.

A. Bit Flip

Bit Flip gate flips the bases state



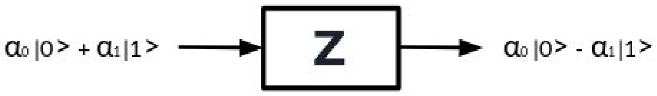
The transformation matrix X is

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

For example,
X |0> = |1>
X |1> = |0>

B. Phase Flip

Phase Flip gate changes the phase of the quantum states.



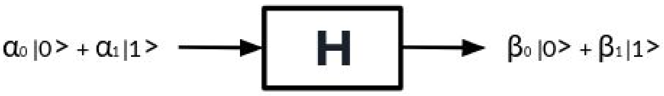
The transformation matrix Z is given by

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

For example,
Z |0> = |0>
Z |1> = -|1>
Z |-> = |+>
Z |+> = |->

C. Hadamard Transform

Hadamard transform gate is one of the most important used in several quantum algorithm circuits.



And the transformation matrix H is

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

For example,
H |0> = |+>
H |1> = |->
H |+> = |0>
H |-> = |1>

Since, the square of gate is identity, i.e. H and H' = **I**. So, if we apply twice, we reach where we start from (initial state).

V. Simon's Algorithm

A. Introduction

Simon's algorithm was conceived to showcase the efficiency quantum algorithms have over classical algorithms. This algorithm provides an exponential speedup over any classical algorithm which uses a black box. In 1994 researchers were debating on how to solve the decision tree complexity problem. At this time Daniel Simon presented a Quantum Algorithms that solves the problem exponentially faster than all other classical algorithms.

Simon's algorithm is the base from which Shor's algorithm is theorized. Both of these algorithms solve the hidden subgroup problem which until recently did not have efficient algorithms.

B. Formulation of the Hidden Subgroup Problem

There are three main steps to this algorithm:

1. Initially for our Quantum Program to run we need a superposition so we can express the data as qubits.

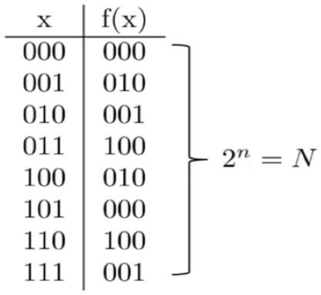
2 Next, we need to perform a fourier sampling on the superposition. This is because a measurement on the superposition itself will not yield any important data, we will simply get a destructive output.

3 Finally, we need to repeat the fourier sampling n-1 times, this allows to get n-1 linearly independent equations. From this linear system of equations we can reconstruct the value of S. In this section we will also explore the probability of getting a linearly independent system.

C. Formulation of the Hidden Subgroup Problem

The hidden subgroup problem is a vast research area in computer science and mathematics. The system comprises of problems like factoring and the shortest vector problem. This problem is particularly important in quantum computing due to its complex nature. Classical Turing Algorithms typically solve these in Polynomial time, whereas algorithms like Simon's and Shor's solves these in linear time.

Let us take a challenge to find a secret string 'S' in a function '$f(x)$' that maps n bit strings to n bit strings such that '$f(x)$ = $f(x+S)$'.



x	f(x)
000	000
001	010
010	001
011	100
100	010
101	000
110	100
111	001

$2^n = N$

Example of Hidden Subgroup Problem

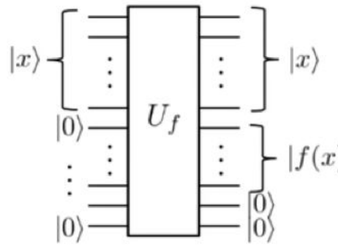
The image above represents our challenge. Here we can see that outcomes of function '$f(x)$' given x for various values of x. Given that n = 3 and the secret string is 'S' = 101. From the image, we can see that without knowing the value of the secret string it is impossible to find a pattern to decode the function '$f(x)$'.

Objective: We are given a 2-1 function '$f(x)$' : $\{0,1\}^n$ = $\{0,1\}^n$ such that there is a secret string 'S' $\{0,1\}^n$ where: '$f(x) = f(x+S)$'. Our objective to find the secret string S.

There is no way to know beforehand the outcome for the inputs to function $f(x)$, hence we know that $f(x)$ is a black box. We will not be able to get a structure for $f(x)$ until we see 2 values for x such that $f(x)$ is the same for both. We will be outlining a quantum algorithm to solve this problem.

D. SuperPosition

Our first step is to represent the equation as a random super-position such as: $\frac{1}{\sqrt{2}}|r> + \frac{1}{\sqrt{2}}|r+S>$. We have '$f(x)$' as a black-box. We also have a quantum program 'U' that computes $f(x)$.

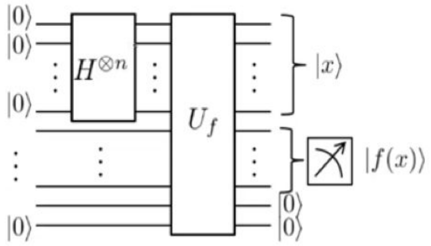


Overview Quantum Computer

Now we need to evaluate the equation as the superposition over x and we get the output as the superposition over x and $f(x)$. The next step is to decide the type of superposition to use.

Let us consider modifying the input using the Hadamard transform. We need to input all the 0 qubit values into the Hadamard transform to get $\Sigma\frac{1}{2^{\frac{n}{2}}}|x>$ which is a uniform superposition over all n bit strings. Now we need to evaluate the output on this superposition to get:

$$\Sigma\frac{1}{2^{\frac{n}{2}}}|x.f(x)>$$



Addition of the Hadamard Gate

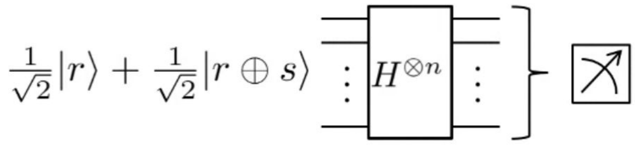
Now we measure $f(x)$, suppose we get some random value 'a'. We know that $f(x)$ is a 2 to 1 function so $f(r)$

= $f(r+S)$ for some random value ‘r’. Now if we measure the string output ‘a’, the first register that gives the superposition over ‘x’, this decomposes to only a few values that result in ‘a’. Finally, we get the superposition as:

$$\frac{1}{\sqrt{2}}|r> + \frac{1}{\sqrt{2}}|r+S> .$$

E. Fourier Transformation

Our next step is to apply the Fourier over the superposition, this is because the measure of the superposition as is would decompose the system. The result of the Fourier transform is: $\sum_y \beta|y>$, where y is a value such that y.s = 0 . (mod 2).


$\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
$H^{\otimes n}$

Initially we assume two cases:

case 1 : y.s = 0 . (mod 2) where $\beta = \frac{1}{2^{\frac{n-1}{2}}}$

case 2 : y.s = 1 . (mod 2) where $\beta = 0$

The result of the Fourier Transform is always a random ‘y’ such that y.s = 0 mod(2).

To show that this is true we will demonstrate a short derivation.

Let us take:

$$\beta = \frac{1}{\sqrt{2}} \frac{-1^{r.y}}{2^{\frac{n}{2}}} + \frac{1}{\sqrt{2}} \frac{-1^{(r+s).y}}{2^{\frac{n}{2}}}$$

Taking the common terms we get:

$$\beta = \frac{-1^{r.y}}{2^{\frac{n+1}{2}}} \ (1 + (-1^{s.y}))$$

If s.y = 1 mod2 the value of β will be ‘zero’ which is destructive to the quantum program. Hence, ‘y’ is always part of case 1.

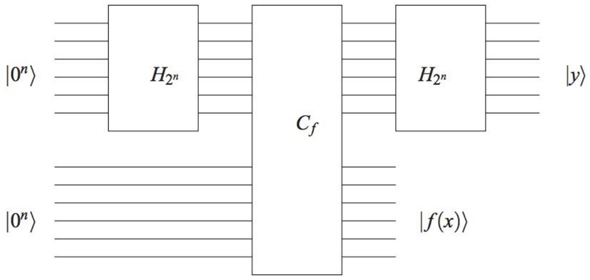


Figure: Representation of SImon’s Algorithm

F. Reconstruction of Secret String S

First, we take n-1 Fourier samples of the super position. We need all the n-1 linear equations to be linearly independent. We could then compute S from those n-1 equations. The objective here is to find the chances of the n-1 equations to be linearly independent.

$y_1s_1 + y_2s_2 + ... y_ns_n$ here, $P(bad) = \frac{1}{2^{n-1}}$

$y_1s_1 + y_2s_2 + ... y_ns_n$ here, $P(bad) = \frac{2}{2^{n-1}}$

.

.

$y_1s_1 + y_2s_2 + ... y_ns_n$ here, $P(bad) = \frac{1}{2}$

The probability of the first linear equation being unusable is only if all the ‘y’ values are 0. From that the probability of computing ‘y’ is $P(y) = \frac{1}{2^{\frac{n}{2}}}$ The second equation is unusable only if ‘y’ = 0 and if the equation is the same as the first one. The probability then becomes $P(y) = \frac{2}{2^{\frac{n}{2}}}$ From this the probability of the entire system to be linearly dependent is:

$$P(bad, n-2) = \frac{1}{2^{n-1}} - \frac{1}{4} \leq \frac{1}{2} + P(bad, n-1) = \frac{1}{2} .$$

Finally, the probability of the system being linearly dependent is $\frac{3}{4}$, so we have a 25% chance to get a linearly independent system from which we can reconstruct S.

G. Conclusion

This algorithm was designed in 1994 to demonstrate the use of quantum computing and is called Simon’s algorithm.

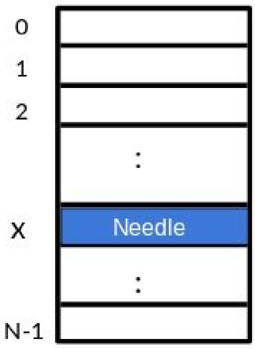
A classical algorithm to solve this black box will typically be in O(2n/2), whereas, a Quantum algorithm can solve this in O(n), this provides an exponential speedup.

VI. Grover's Algorithm

A. Introduction

Now, we'll talk about another monumental invention in the field of Quantum Computing, which is Grover's Algorithm. Lov Kumar Grover came up with this in 1999, while working at Bell Labs. It's equivalent to searching a needle in haystack. Grover algorithm performs searching in unordered data. By using classical methods, it takes out O($\frac{N}{2}$) average case, by trying out random entries and worst case of O(N).

So, now how we can improve it using exponential power of quantum mechanics. So, we are hoping to find some way, such that we don't need to search for item and use something like say a magnet to pull needle out of the haystack. But, it's not going to be that fast that it'll finish in single step. But, nonetheless very clever algorithm.



Before going forward, let's see why this problem is so important. As, we know there's a whole class of problem, NP-complete problems. Let us pick one of them say satisfiability problem(SAT), which ask to find assignment to f(x_1, x_2, ... x_n) which solves the given formula. There are 2^n possible configurations.

Now, we can take it as search algorithm with say size N table where N = 2^n and each row contains one of the value of f(x). Similarly, there are lot more which falls in same NP complete problem and if we can improve this problem, they'll improve as well.

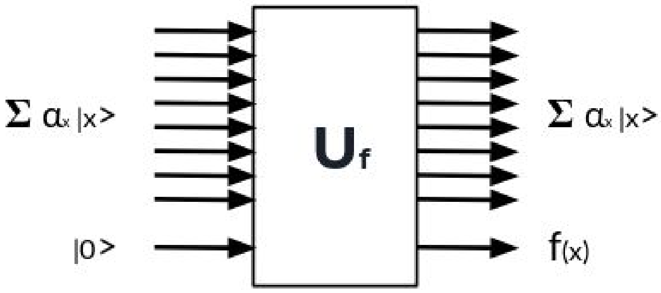
Coming back to grover's algorithm. So, grover algorithm can solve the problem in $\sqrt{N}$ time, which is quadratic speedup from the classical solution. Even though, it gives very significant speedup, but for SAT problem it's still leaves with O($2^{\frac{n}{2}}$) i.e. exponential time algorithm.

B. Formulation of Grover's Algorithm

Let's see the Grover Algorithm

Problem: Given f: { 0, 1, ... N-1 } → { 0, 1 }
Find x : f(x) = 1

We'll use the quantum circuit to solve the above formulated problem. We can define the quantum circuit which takes input set of x and returns f(x). So, the whole point here is we can evaluate x and f(x) in superposition.



So,

$$\sum_x \alpha_x |x>|0> \Rightarrow \sum_x \alpha_x |x> f(x)$$

So, now let's discuss the steps of Grover's algorithm and then see how to implement those steps. There are two major steps to the problem. First is called Phase Inversion. Second step is called Inversion about the Mean. Let's understand first what they mean before jumping into Grover's algorithm step implementation details.

C. Phase Inversion

Let's assume the special entry is x*, so the f(x*) = 1. So, in any iteration of algorithm, it maintains superposition over all the other x. Initially, we'll have all the amplitude to same value (after being Initialization step) and hence value will be $\frac{1}{\sqrt{N}}$.

So, in phase inversion step, we change the superposition such that if x$\neq$ x*, then leave it as it is. But, if it's x = x*, then we change the phase. We can formulate it as following

$$\sum_x \alpha_x |x> \Rightarrow \sum_{x \neq x*} \alpha_x |x> - \alpha_{x*} |x*>$$

D. Inversion about the Mean (μ)

Second operation in Grover's algorithm is inversion about the mean. Again we start with superposition

$$\sum_x \alpha_x |x>$$

and let's assume μ be the mean, which will be

$$\mu = \frac{\sum_{x=0}^{N-1} \alpha_x}{N}$$

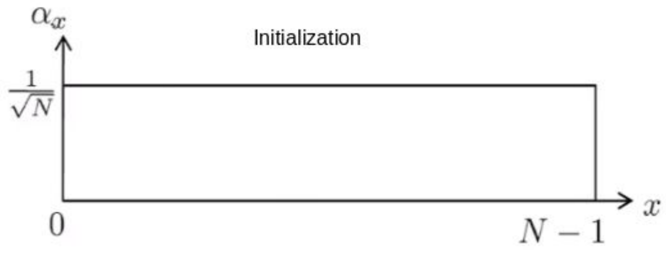
So, now we flip the amplitude about this mean and we map α_x to $2\mu - \alpha_x$, which means

$$\sum_x \alpha_x \,|x> \quad \Rightarrow \quad \sum_x 2\mu - \alpha_x \,|x>$$

But, why $2\mu - \alpha_x$? Since, $2\mu - \alpha_x = \mu + (\mu - \alpha_x)$
So, if $\alpha_x < \mu$, it'll be smaller by $\mu - \alpha_x$. Hence, flipping it'll be $\mu + (\mu - \alpha_x)$. Similarly, it works other way as well when $\alpha_x > \mu$.

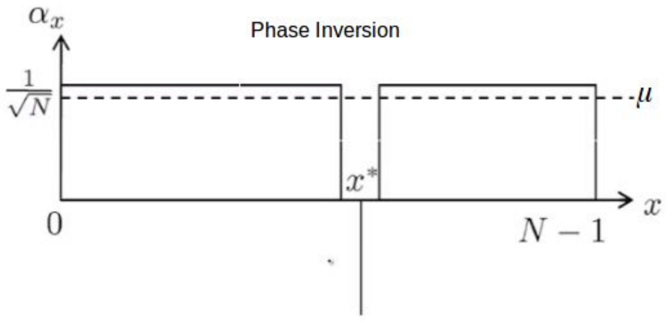
E. Steps for Quantum Search

Now, as we have understood basic concepts needed for operations needed for Grover's algorithm. Let's see how we use these in algorithm implementation. We start with amplitude for all the superposition of X to be equal to $\frac{1}{\sqrt{N}}$



After the iteration, we do the phase inversion for the marked element's amplitude. Making its amplitude $-\frac{1}{\sqrt{N}}$ and then we do inversion about the mean. Now since one of the element (Marked element, x*) went $\frac{1}{\sqrt{N}}$ => $-\frac{1}{\sqrt{N}}$

So, mean (μ) will be little less than $\frac{1}{\sqrt{N}}$.

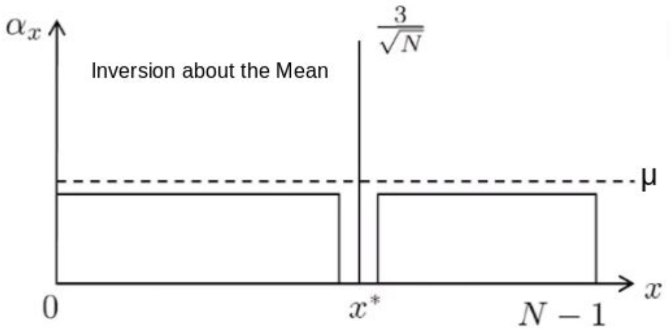


Now, after the Phase inversion, when we do the Inversion about the Mean, the amplitude of x* will be $\frac{3}{\sqrt{N}}$.

Since, $\mu \simeq \frac{1}{\sqrt{N}}$.

So, the
$2\mu - \alpha_{x*} \simeq 2.\frac{1}{\sqrt{N}} - (-\frac{1}{\sqrt{N}})$

$2\mu - \alpha_{x*} \simeq \frac{3}{\sqrt{N}}$



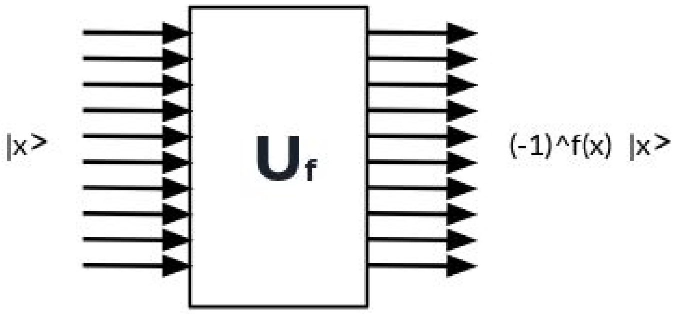
So, now we keep doing this step over and over and over. After the next iteration we get the amplitude of $\frac{5}{\sqrt{N}}$ of x* and we get increase of about $\frac{2}{\sqrt{N}}$ approximately. So, it goes to $\frac{3}{\sqrt{N}} \rightarrow \frac{5}{\sqrt{N}} \rightarrow \frac{7}{\sqrt{N}} \rightarrow \frac{9}{\sqrt{N}} \rightarrow ... \rightarrow \frac{1}{\sqrt{2}}$ in $\sqrt{N}$ steps (iterations) and hance in $\sqrt{N}$ iterations we get the probability of x* superposition of $\frac{1}{2}$ $(|\alpha_{x*}|^2)$.

F. Quantum Circuit

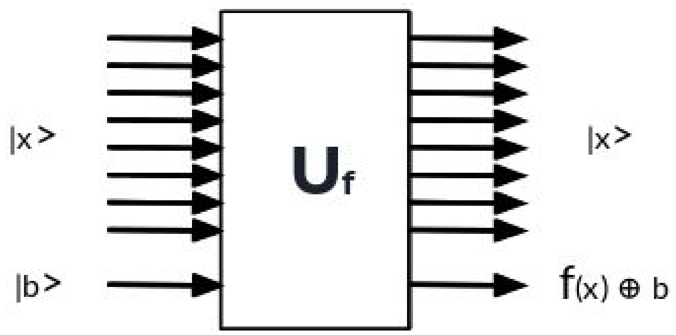
After seeing the steps, let's see how we perform these steps using quantum circuit.
For Phase inversion, we'll achieve following transition

$$\sum_x \alpha_x \,|x> \quad \Rightarrow \quad \sum_x \alpha_x(-1)^{f(x)} \,|x>$$

So, we'll need gate like



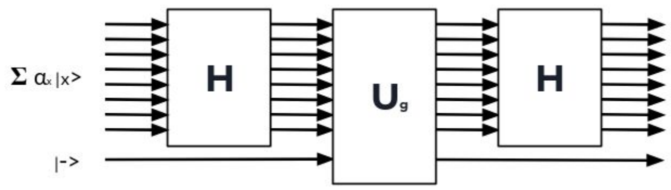
We can do that by using following circuit,



Now for circuit to perform Inversion about the mean, for this we want transformation like below,

$$\sum_x \alpha_x |x> \;\Rightarrow\; \sum_x (2\mu - \alpha_x) |x>$$

We can achieve using following circuit,

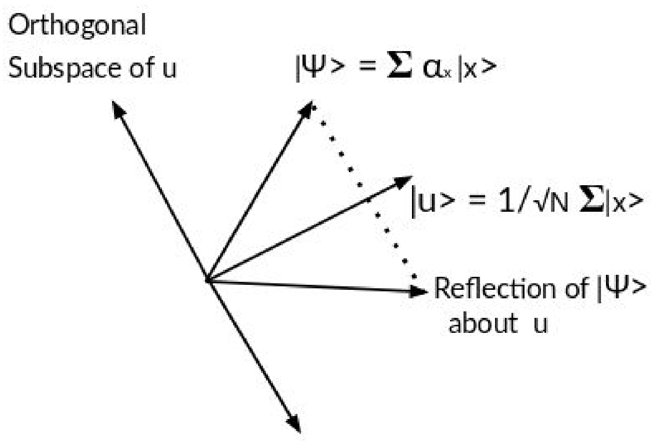


In the about circuit, for the gate U_g , we assume if all previous input in y (output from H). Then if we input 0 for the last input, output will be

$$\begin{array}{l} 0 \text{ if } y = 0...0 \\ 1 \text{ if } y \neq 0...0 \end{array}$$

Again we do the Hadamard transformation (H) after this. In fact what we are doing in 'Inversion about the mean' circuit is just the reflection about the mean superposition state $|u>$

$$|u> = \frac{1}{\sqrt{N}} \sum_x |x>$$



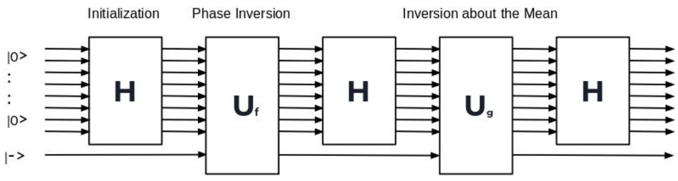
So, to perform above reflection we need to following steps:

1. First, we transform the $|u>$ to all 0 vector |0….0>, which is performed by the first Hadamard gate (H)

2. Next we perform the reflection about |0….0>, which is achieved by the gate U_g . During this reflection we basically leave all zero vector alone and everything that is orthogonal to it gets multiplied by -1.

3. Finally, we transform the all 0 vector |0….0> back to $|u>$. This step is performed using the second Hadamard gate (H) in the above quantum circuit.

Once we assemble the all these steps, we get following circuit which implements end to end the Grover's Algorithm.



Repeating this √N iterations

G. Conclusion

So, we understood Grover's algorithm actually finds the desired unique input using it's high probability for a black box function that produces a specific output value, using just about $\sqrt{N}$ iterations of the function. After in 1996, Grover designed the algorithm, Vazirani, Bennett, Bernstein and Brassard proved that no other quantum solution to the problem can run the function in less than O($\sqrt{N}$). Hence, the Grover's algorithm is technically asymptotically optimal.

But, this talks about the approach used by Grover is optimal in quantum evaluation. As the growth is happening in the quantum computation field, I wouldn't be amazed if I read tomorrow about another approach to solve similar problem in optimal time. But, nonetheless it was significant speedup from the classical counterparts by Lov Grover. Actually, even the quadratic speedup is very huge in cases where N is very large number. For example, in case to attempt break a 256-bit cryptographic key using brute-force. Using the classical systems, we'll need 2^{256} attempts. But, whereas using Grover, it gets reduced to 2^{128} iterations.

VII. Contribution

As team of two students, we split the work equally. We first started by studying the basics of quantum mechanics. We investigated different lectures and publications and taught each other about how quantum mechanics is used to achieve computation and building algorithms using that. Then after we were clear about how quantum computation works, we picked one of our favourite algorithms and start working and reading about them.

We decided to investigate and explain Simon's Black-Box algorithm and Grover's Quantum Search Algorithm. I worked on Grover's Algorithm and Laksh worked on Simon's Algorithm. We read the different publications and saw video lectures by famous physicists to understand deeper about some of the concepts as we were discovering them. At the final step of the project, we realize now that we have learnt a lot about quantum computation, but still there is lot to learn, and hopefully during this journey, sometime in future we might make some contributions to this field of computing.

VIII. Conclusion

From our explanations of Simon's and Grover's algorithms above we can see the raw power Quantum algorithms have over classical Turing systems. An exponential and quadratic speed respectively provides an incredible boost and reduced the time complexity by a large order of magnitude. Now we will exhibit some key real world uses and caveats of Quantum Algorithms.

Online cryptography will be forever changed after the advent of modern Quantum Computers. Current cryptography methods will become obsolete. Most traditional cryptography methods rely on the feature that it takes a very long time to crack the code. This will be rendered useless as Quantum Computers will be able to perform arithmetic on very large numbers in a small matter of time. The silver lining here is that a lot of work is being done with respect to protection methods using Quantum Algorithms. One such example is Quantum Key distribution, which requires a key to decode packets, if this packet is intercepted it will disintegrate.

Another key use will be with machine learning. Data processing takes a lot of time for machine learning, it can be easily cut down using Quantum Computers. Quantum Algorithms can help us process large amounts of data to give us insights in a much more timely fashion. Quantum Computers can process training data much faster than traditional systems, this allows us to reduce the training time for machine learning algorithms.

Drug development is a very complex task, having chemists run interactions with proteins, molecules and other vital ingredients. From this we can see that there are hundreds of thousands of possible combinations to run through. This involves a lot of effort and takes a lot of time. Quantum Algorithms in this case can help us review multiple molecule, proteins and other ingredients in a short period of time. Quantum Computers can also help sequence genes in a fraction of the time traditional systems take.

IX. References

1. David Deutsch's Lectures on Quantum Computation
   http://quiprocone.org/Protected/DD_lectures.htm
2. Umesh Vajirani's Lectures on Quantum Mechanics and Quantum Computing
   https://www.youtube.com/watch?v=OkkAwRgcnWA
3. Timeline of Quantum Computing
   https://en.wikipedia.org/wiki/Timeline_of_quantum_computing
4. Quantum Frontiers (CalTech)
   https://quantumfrontiers.com/2013/08/22/the-most-awesome-animation-about-quantum-computers-you-will-ever-see
5. Unboxing a Quantum Computer - DWave
   https://www.youtube.com/watch?v=60OkanvToFI
6. Grover's Algorithm
   https://en.wikipedia.org/wiki/Grover's_algorithm
7. Algorithms implemented by 1QBit
   https://1qbit.com/technology/
8. IBM QuBit Simulator
   https://quantumexperience.ng.bluemix.net/qx/editor
9. Simon's Problem
   https://en.wikipedia.org/wiki/Simon%27s_problem
10. Uses of Simon's algorithm
   https://courses.edx.org/c4x/BerkeleyX/CS191x/asset/chap4.pdf
11. Explanation of Simon's algorithm
   https://cs.stackexchange.com/questions/19519/simple-explanation-of-simons-problem

X. Acknowledgement

We would like to sincerely thank Professor Ashutosh Trivedi for his generous contributions to this project. We would also like to thank the CU Boulder Computer Science Department for providing us the opportunity to pursue and present this work.