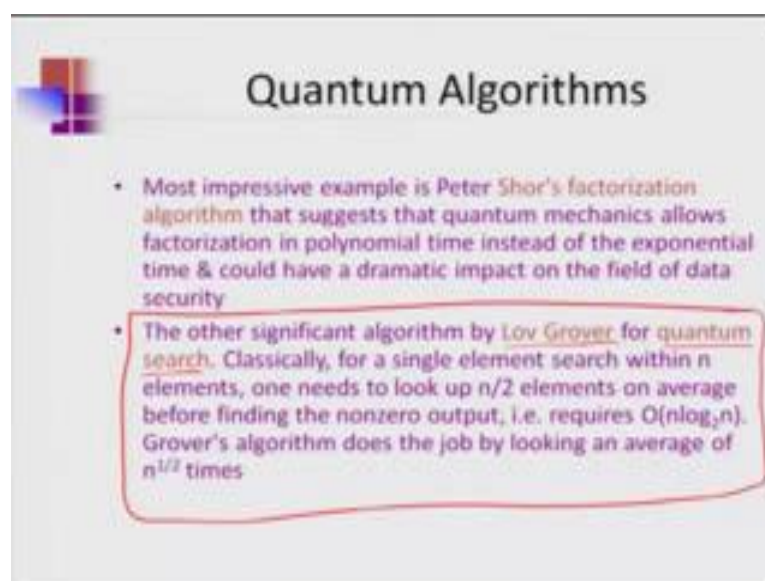


Implementation Aspects of Quantum Computing
Prof. Debabrata Goswami
Department of Chemistry
Indian Institute of Technology, Kanpur

Lecture – 08
Grover's Algorithm

Today we are going to discuss about quantum algorithms.

(Refer Slide Time: 00:19)



Quantum Algorithms

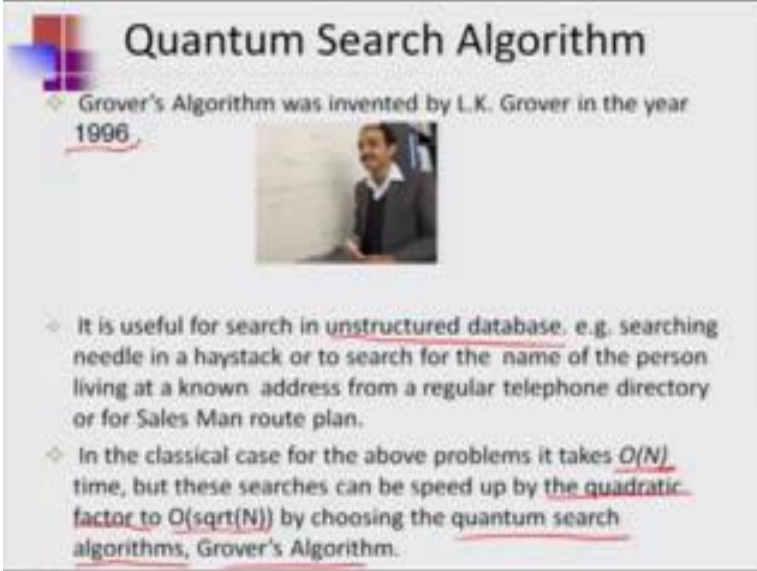
- Most impressive example is Peter Shor's factorization algorithm that suggests that quantum mechanics allows factorization in polynomial time instead of the exponential time & could have a dramatic impact on the field of data security
- The other significant algorithm by Lov Grover for quantum search. Classically, for a single element search within n elements, one needs to look up $n/2$ elements on average before finding the nonzero output, i.e. requires $O(n \log_2 n)$. Grover's algorithm does the job by looking an average of $n^{1/2}$ times

The most impressive example is Peter Shor's factorization algorithm that suggests that quantum mechanics allows factorization in polynomial time instead of the exponential time and could have a dramatic impact on the field of data security.

This is the most impressive one. However, this takes a little bit more of understanding so we will deal with it later the other significant algorithm is the one which will take up first today is more known as the search algorithm. This significant algorithm was developed by Lov Grover for quantum search. Classically, for a single elements search within n elements one needs to look up n by 2 elements on an average before finding the nonzero output. That is requires an order $n \log_2 n$, but the Grover's algorithm does this job by looking at an average of n to the power half basically root n times.


So, this advantage that is achieved by quantum search is something that we will discuss today. In order to get quantum algorithms what are the significant steps. Some of the basic ideas we have already discussed, so we will see how this one involves.

(Refer Slide Time: 01:46)



Quantum Search Algorithm

Grover's Algorithm was invented by L.K. Grover in the year 1996.



- It is useful for search in unstructured database. e.g. searching needle in a haystack or to search for the name of the person living at a known address from a regular telephone directory or for Sales Man route plan.
- In the classical case for the above problems it takes $O(N)$ time, but these searches can be speed up by the quadratic factor to $O(\sqrt{N})$ by choosing the quantum search algorithms, Grover's Algorithm.

Grover's algorithm was invented by Lov Grover in the year 1996. It is useful for search in unstructured data base; that is in the case where the data set is not arranged in the way so that you can look after one of the other. Because if you have an arranged data set then the way of searching it can be done much faster by using classical structures.

However, for example to search a needle in haystack or to search the name of a person living in a known address in a regular telephone directory. These are very hard problems, and this can be done by using this technique. It is also applicable for sales man to find out a route his plan for selling items, all these kinds of applications this particular problem is applicable. In the classical case for any of these above problems it takes an order n as time, but these searches can be speed up by the quadratic factor to orders square root of n by choosing the quantum search algorithms which is also known as the Grover's Algorithm.

The basic idea behind this algorithm is to come up with a technique which is not going to look at the data set one by one, but it is going to do it in a quantum way so that it is done at a much faster way, that is the basic idea.

(Refer Slide Time: 03:14)

The Oracle

- ❖ Oracles have the ability to recognize solutions to the search problem. The recognition is signaled by making use of an oracle qubit.
- ❖ Oracle is an unitary operator, O , defined as :

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle$$

$|q\rangle$ is a single oracle qubit which is flipped if $f(x)=1$ and is unchanged otherwise.

- ❖ In the equation given below, state of the oracle qubit is unchanged, which is easily observable

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{O} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

The oracles have the ability to recognize solutions in the search problem. The recognition is signaled by making the use of an oracle qubit in this particular case because it is going to be quantum case. So, the oracle will be a unitary operation which can be defined as an operation of the qubit on, so defined as the operator O which works on the set that we are looking at such that we create a functional dependence of the two and where q is a single oracle qubit which is lift if the function is equal to 1 and it is unchanged otherwise.

So, basically the set up of the oracle as an unitary operator can be defined in this particular way, where by using the single qubit q which is our oracle qubit the idea is to make it flip when the function changes; that is when I have found the solution the such that we are looking for or it remains unchanged. So, as long as the operator does not give the result of being flipped we continued in the process, we look for the search.

In the equation given below the state of the oracle qubit is unchanged which is easily observable. For example, when you take a state like this and we finally see what is going on with it the oracle qubit keeps remaining the same which means that I can keep on using it over and over again. So, that is the idea behind an oracle that the oracle should keep on reporting the right solution in spite of not changing.

(Refer Slide Time: 05:18)

The Oracle (contd...)

- So the oracle can be written in simplified form as:
$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$$
- Oracle doesn't know the solution but can recognize it, and against our common intuition it is possible to do the latter without doing the former.

Example: Deutsch Algorithm studied before

Oracle

A black box that can recognize the solution, whose internal working is represented by a binary function $f(x)$

$$f(x) = \begin{cases} 0, & (x = \beta) \\ 1, & (x \neq \beta) \end{cases}$$

$x \rightarrow$ Oracle \rightarrow 0 or 1

In a simplified form therefore, the oracle can be written in this form where if you have the qubit then you are going to have the function which is going to have this minus 1 f of x to the power f of x times the qubit that we are looking for.

In other words the oracle does not know the solution, but can recognize it and against our common intuition it is possible to do the latter without changing the former. The most important part here to remember is that it does not change the oracle bit. An example of that you already have we have seen it before, in case of the Deutsch Algorithm where the black box which we call the oracle is the one which can recognize the solution whose internal working is represented by a binary function. Once we put it through the oracle the functional form would be giving out either 0 or 1 although the oracle keeps it the same way.

So, basically I can keep using the oracle over and over again and get the algorithm go. And this is one of the ideas of making sure that we can go ahead and apply quantum concepts to develop a faster way of doing the computing.

(Refer Slide Time: 06:42)

Grover's Iteration

- ◇ The quantum search algorithm consists of repeated application of a quantum subroutine known as Grover iteration or Grover operator.
- ◇ Steps of the Grover iteration :
 - 1) Apply the oracle O .
 - 2) Apply the Hadamard transform.
 - 3) Perform a conditional phase shift on the computer, with every computational basis state except $|0\rangle$ receiving a phase shift of -1 .
 $|x\rangle \rightarrow -(-1)^{f(x)}|x\rangle$
 - 4) Again apply the Hadamard transform.

mixes about mean

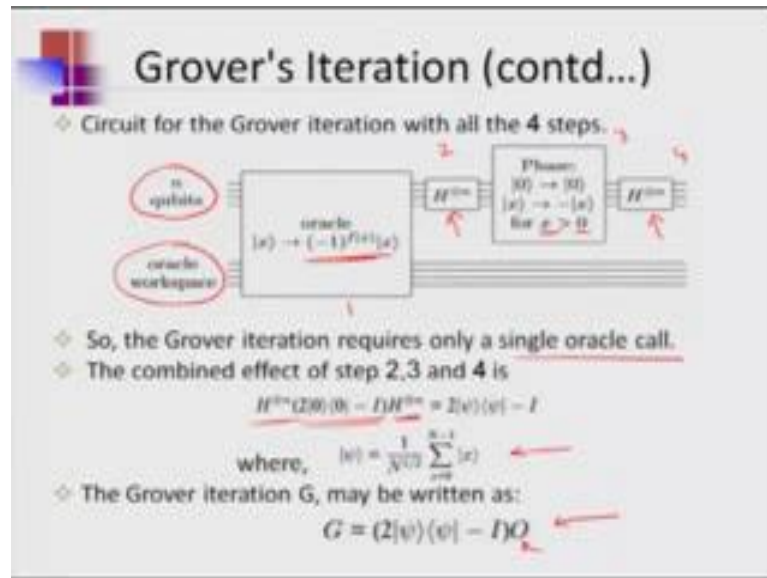
So, now given this background it is understood that there will be several iterations that will be necessary to get the solution from the search algorithm. So, the quantum search algorithm consists of repeated application of a quantum subroutine known as Grover's iteration or Grover operator. In the classical sense when we were suppose to search, we were suppose to go through each and every data point to be able to get to the solution. In this particular case it consists of repeated application of the quantum subroutine instead of having to look through each and every data; that is the basic idea.

In other words the steps of the Grover at iteration are; number one is to apply the oracle O . So, first of all you have to set up an oracle as we discussed in such a way that it can recognize what we are looking for the search. Then we need to apply the oracle, apply the Hadamard transform. Now Hadamard transform is something which we have applied before, which has learnt about before. This transformation essentially takes the qubits and mixes them up. So, if you take two qubit situations then it will give equal weightage and put them together, that is the idea of Hadamard transform.

So, we will again revisit that to iterate our memory, but generally this is how the transform is. Then we will do a conditional phase shift on the computer with every computational basis state except the 0 receiving a phase shift of minus 1 . So, that is how we will take our data set from x to x with this particular functional dependence where we will get this. So, this is essentially inversion about me.

So, in essence this conditional phase shift on the computer which every computational basis state except 0 receives a phase shift of this is almost like inversion about the mean of the entire data state; that is what we will do. And then again we will apply the Hadamard transforms so that we can get back to the data state as we had before and be able to understand what we have just done. So these are the steps.

(Refer Slide Time: 09:19)



In other words if you want to write it in terms of the circuit; circuit for the Grover iteration with all the four steps that we just mentioned looks like this. We have the n qubits which need to be loaded up, and then we have the oracle works phase that we have just discussed. Once we put this together inside the oracle it is essentially undergoing a transform where the functional decision as to whether the function value is 1 or not is being found so that you apply the oracle and get the solution. And then you again do a Hadamard transform, apply the phase shift around the mean about 0 it remains the same, it means that it is actually doing an inversion about the mean. And then apply the Hadamard transform again.

In some sense therefore, the Grover's iteration requires only a single oracle call, because once you are set it up put it in the oracle work space, you just lead to set it up once to be able to find out whether you have got the solution or not. The combined effect of steps 2 3 and 4; so 2 3 and 4, so this was our first step is that we have applied a Hadamard on this initial set and a reverse Hadamard and depending on if you at doing it once or that

many times we will have it to the this operation applied that many times so that we are able to get the solution. And our wave function therefore, is represented by this normalized case $\frac{1}{\sqrt{n}}$ to the power half \times vector.

So, the Grover iteration G then therefore is written as this particular set G is equal to 2 times wave function put in to different ways minus i and then the operation that we are applying which is the oracle operation. So, this is the iteration process that will continue. And once we keep on doing this iteration we will be finally getting the solution.

So, this is the iterative way of doing this particular approach of Grover's algorithm. Let see how it look.

(Refer Slide Time: 12:15)

Geometric Visualization

- ❖ Grover Iteration can be regarded as a rotation in the 2D space.
- ❖ Σ^- indicates sum over all x which are solutions to the search problem and Σ^+ is the sum over all x which are not solution to the search problem.

$$|w\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \in \Sigma^+} |x\rangle$$

$$|v\rangle = \frac{1}{\sqrt{M}} \sum_{x \in \Sigma^-} |x\rangle$$

- ❖ Initial state may be re-expressed as;

$$|w\rangle = \sqrt{\frac{N-M}{N}} |w\rangle + \sqrt{\frac{M}{N}} |v\rangle$$

- ❖ Oracle operator performs a reflection of about the vector, such that: $O(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle$

We can also visualize this geometrically because, many a times this entire process can be understood much better if we visualize this geometrically. So, this can in that sense we regarded as a rotation in the 2D space, and that is a perhaps a much more intuitive way of understanding how this is going. In this process we can we can actually look at this entire process again in the same way. Maybe I should show the figure before I go ahead with this.

(Refer Slide Time: 13:54)

Geometric Visualization

- Similarly the operation performs a reflection about O, and the product of these two reflection is a rotation.

Let $\cos \theta/2 = \sqrt{(N-M)/N}$
 so that $|y\rangle = \cos \theta/2 |x\rangle + \sin \theta/2 |z\rangle$

$G|y\rangle = \cos \frac{3\theta}{2} |x\rangle + \sin \frac{3\theta}{2} |z\rangle$

$G^2|y\rangle = \cos \left(\frac{2\theta + 1}{2}\theta\right) |x\rangle + \sin \left(\frac{2\theta + 1}{2}\theta\right) |z\rangle$

So, here is the geometric visualization where we have this 2D access and this is my state that we are looking at, so this is my wave function. And depending on whether I am rotating it which whichever way. So, this is my state alpha, this is my state beta these are the two components of the wave function that I have. Now my operation can essentially take it down one way where the other and around one of the mean positions, so if alpha is my mean position I can have this go down by theta by 2 or theta by 2 in either direction or we can go the entire direction by theta.

So, basically it is a reflection about the wave function that we can go in either direction. The operation performs a reflection about O, and the product of these two reflections is a rotation. So this is my operation which is essentially doing the reflection and the product of these two reflections is a rotation. Let me again go back to the earlier slide to essentially indicate all the mathematic associated with this. The sum over all the x which are the solutions to the search problem and double prime, say summation prime of x and summation double prime of x at the two sums one of them is the one which is solutions of the search problem and the other one double prime one is the sum over all x which are not solution to the search problem.

So, there are two cases; basically one which is the case with the solution and the case which is without the solution. Then we can set up two different states say alpha and beta. This is what I will trying to represent when I went back to the other picture. So, we will

go back to this after setting it up now. See alpha and beta are the two cases where we have set up between solutions versus no solution. So, the initial state may be represented as something which is super position of these two conditions where we have the solution and we do not have the solution. And that is the way how you can read it.

And then the oracle operator performs a reflection about the vector such that it will route it such that we will be able to find the description of the state amongst that between the solution case and the one which does not have the solution case, and based on these we can get our solution. So we have a solution case; with solution is this one and without solution is the one where the alpha.

So, now my wave function is somewhere in between alpha and beta which basically means that it is a super position of the two or having contribution from both of them. See if we consider an angle theta such that the cosine theta by 2 is the weightage factor n minus m over n root over square root then we can tell that the wave function is cosine theta by 2 alpha plus sin theta by 2 alpha beta.

So, that is how we have set up the wave function. Now the Grover application in the state essentially takes it $2 \cos^3 \theta$ by $2 \sin^3 \theta$ by 2 which is where this is the rotation we are getting. If you apply it k times then this is essentially be a rotation which will bring it back and forth around that point all the time. So that is the idea behind applying the Grover's algorithm over and over again.

In fact, this also tells you one very interesting thing that there is a point and this will do it later in a more detail fashion, where we will show that there is a point beyond which the solution finding essentially oscillates around the actual signal. So, you find out the solution by using Grover's algorithm. And if you keep on running the iteration there is nothing like further improvement or something it essentially oscillates back and forth around the actual solution.

So, once you have gotten your solution you cannot really just say that you know you can continue on doing it to get to anything better, it will just be oscillating around the actual solution; that is the idea.

(Refer Slide Time: 18:00)

Algorithm

- The number of times Grover iteration has to be applied has an upper bound given by : $R \leq \lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \rceil$
- i.e $R = O(\sqrt{\frac{N}{M}})$ Grover iteration must be performed in order to obtain a solution to the search problem.

Algorithm: Quantum search

Input: (1) A black box oracle O which performs the transformation $O|x\rangle = (-1)^{f(x)}|x\rangle$, where $f(x) = 1$ for all $x \in S$ and $f(x) = 0$ for all $x \notin S$. (2) A list of N items $|x\rangle$.

Output: $x \in S$ with probability $\geq \frac{1}{2}$.

Procedure:

- Initialize $|x\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ (uniform superposition)
- Apply O to $|x\rangle$ (oracle)
- Apply A to $|x\rangle$ (diffusion operator)
- Repeat steps 2 and 3 for R iterations.
- Measure $|x\rangle$ to obtain $x \in S$.

So anyway, the number of times the Grover's iteration has to be applied has an upper bound therefore, beyond which it essentially means that it is just going to oscillate. And that is given by this function that is given by this R which is basically coming from the equations that I just wrote in the last slide. So, if you apply this, how many times you apply and you get number beyond which it has no meaning. So that is the value that you can actually come down to pi over 2 square root of N over M. That means, that my R has the order of square root of N over M. Grover situation must be performed in order to obtain a solution to the search problem and that takes an order which is square root of N over M.

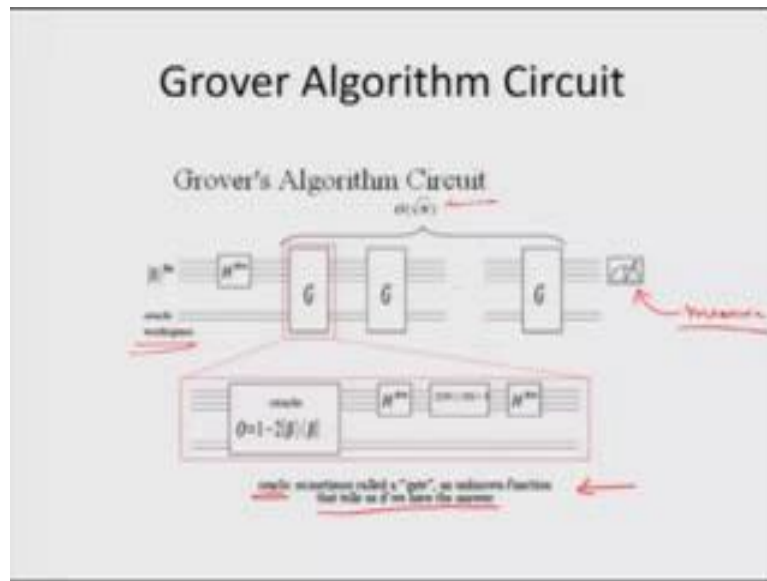
So, the basic idea behind this quantum search is that we are using a black box oracle O which performs the transformation as given in this function, such that when the function has some values except the one with this it is going to set up the value of f x is 1 otherwise it will have the other answer f x equal to 0 and there are n plus 1 qubits in this state which is 0. The output is going to be the function the x 0 which is your solution and the run time will be of order in this particular case order 2 to the power n operations. It succeeds with probability of order 1 which means it is almost always it is going to be successful it just that there is a time bound beyond which it just keeps on oscillating.

So, the procedure involves a setting up the initial state, then applying Hadamard transform on the first time qubits and h x to the last qubit which is the result of using the

on the last qubit, x is the not gate as we have said. And then we apply the Grover's iteration with this 2 to the power N over M . We have basically chosen the special case where our N by M has is of the order 2 to the power N .

So, we get this and then once we get the solution which is of this order this is our oracle qubit so we preserve that. And our solution is $x 0$ which is the first and qubits that we are looking for. So, that is the basic algorithm behind the Grover's case.

(Refer Slide Time: 12:04)



And here is how this circuit looks like, this particular case often is also known as it is an unknown function which tells us if we have a function. So, this is the oracle which often sometimes or is also known as the Grover gate. In some cases it is an unknown function which tells us if you have an answer. So, this is the Grover application of the Grover algorithm looks like the G gate or something.

And this we apply it on the on the order of root N times and we should be able to get our proper solution and this is our measurement. As we know this is what we have setup is our measurement instead. And this oracle work space or work space always maintains oracle qubit which does not change as a result of this process.

(Refer Slide Time: 22:08)

Optimality

- ❖ Classical algorithms take $O(N)$ operations for searching N items.
- ❖ Grover's algorithm can search N items by calling the oracle only $O(\sqrt{N})$ times.
- ❖ It is shown that Grover's algorithm is optimal, i.e., any quantum algorithms require at least $O(\sqrt{N})$ times oracle calling for searching.

Unstructured Data set
not ordered

So, the one very important thing to note here as a result out of this is the optimality of this process. The classical algorithm take order N operations for searching N items, Grover's algorithm however can search n items by calling the oracle only square root of N times and it is at least quadratic speed up. It is also shown that Grover's algorithm is optimal that is any quantum algorithms requiring at least square root of N times would be oracle calling for searching.

And, one of the important things to mention here that this entire picture is essentially true when it is an unstructured data set. This is because if you have already got a data set which is sought out structured in the sense that it is ordered. If it is ordered in some way say from higher to lower numbers something like that then the this idea of benefit from the Grover search does not exist. Also because in that case the operational procedures required to find the solution is of a different kind.

For instance, in case of classical case if you have the prior knowledge that this data set is structured in some way you will not have to look through the entire data set. And therefore, the optimality or re-advantage that is required or necessary and which comes from Grover's algorithm is absent, and therefore there is no benefit in the absence when we use the other kind of data set.

So, this particular algorithm is usually effective in the case of unstructured data set. So, that is the main issue.

(Refer Slide Time: 24:12)

Inversion about Mean

$$\begin{aligned}
 & \langle \mathbb{I} \psi | \psi \rangle = \sum_k a_k \langle k | k \rangle \\
 & = \sum_k a_k \langle k | k \rangle = \sum_k a_k \langle k | k \rangle \\
 & = \sum_k a_k \langle k | k \rangle = \sum_k a_k \langle k | k \rangle \\
 & = \sum_k (2\langle \alpha \rangle - a_k) \langle k | k \rangle
 \end{aligned}$$

$$\begin{aligned}
 \langle \psi | &= \sum_k a_k \langle k | \\
 \langle k | k \rangle &= \delta_{kk} \\
 \langle \alpha \rangle &= \sum_k a_k \langle k | k \rangle \\
 k' &\rightarrow k
 \end{aligned}$$

And a small note about this other step that I have been talking about which is the setting the value in such a way that everyone around the 0 is going to change their sign, where the one at this at the 0 remains the same and that is where we have mentioned that the values go to minus the case when it is about the 0 values otherwise it remains the same.

So, this is the inversion about mean. For any data set that is one of the steps that we have used. What we have done is that, around the mean position which for example is this case alpha any number which is around the alpha mode alpha or the mean value it will be changing its value, whereas if it is on one side of it it will remain the same. So, that is the idea about inversion about the mean which is also applied in this case, very effectively to be able to mark out the right point.

So, what we have done in today's class is we have gone through one of the very important algorithms which was discovered by Lov Grover for quantum search and it turns out to have at least a quadratic improvement over the classical search algorithms. And, what we expect to do is to see a lot more of this algorithm, because search turns out to be a very important part of any of the computational problems, because we can always assume that you will have an answer which we will have to look for whenever you are doing your problem.

So, that way this is a much more general problem to look for. And in most cases in those cases you do not have any structural knowledge essentially an unsorted set that you are

looking through to find your solution. And therefore, Grover search happens to be a very useful set of search algorithm which has an immense benefit once you go to quantum approach.

So, this is the one which you have done. The other one which is the Shor's factorization we will take it up on a later case. Most likely we will look in to some more aspects of Grover's algorithm in the next class where we will perhaps look at its implementation or its other features as we get into the next lesson.

Thank you.