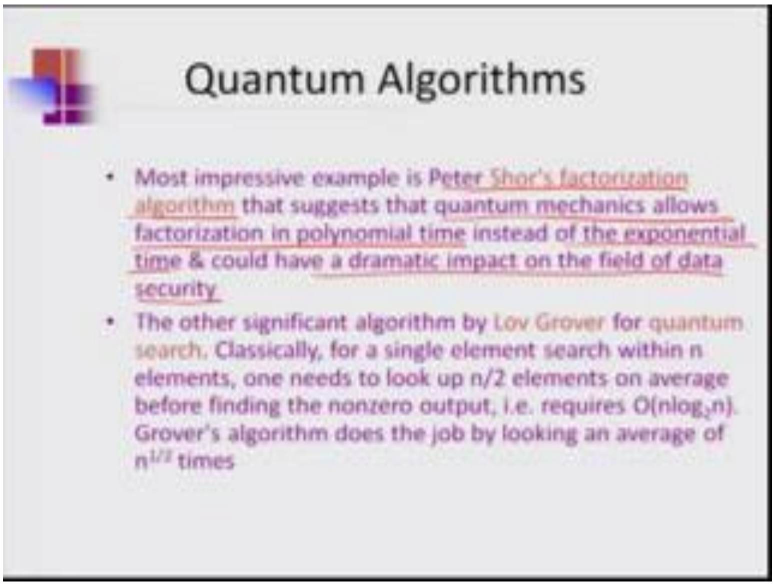
Implementation Aspects of Quantum Computing
Prof. Debabrata Goswami
Department of Chemistry
Indian Institute of Technology, Kanpur

Lecture - 9
Basics of Shor’s Algorithm

We are going to look into Quantum Algorithms again, but this time we will be looking at the one which is the most impressive example - the Peter Shor's factorization algorithm.

(Refer Slide Time: 00:19)



Quantum Algorithms

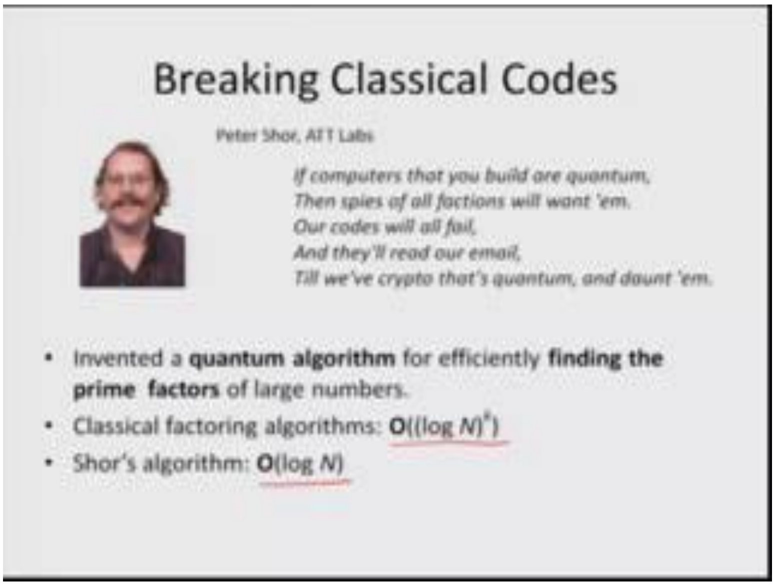
- Most impressive example is Peter Shor's factorization algorithm that suggests that quantum mechanics allows factorization in polynomial time instead of the exponential time & could have a dramatic impact on the field of data security
- The other significant algorithm by Lov Grover for quantum search. 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

This 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 now the reason why this is so amazing. Is if you can get factorization done in a time frame, which is in polynomial as compared to the present exponential time and dependent methods classically available. Then it would mean that the data security, which is basically based on the idea prime factorization with the large number is going to have a lot of difficulty in maintaining security.

That is why this particular problem is very interesting to look at on one side it also provides on the other side better ways of creating cryptography, because if you can do this kind of exponential benefit then you can come with better aspects of cryptography which can be applied. This has big advantage and understanding of the security issues.

So, Peter Shor's who is at the ATT bell labs, came up with this idea in the once again in the 90s.

(Refer Slide Time: 01:42)



Breaking Classical Codes

Peter Shor, ATT Labs

If computers that you build are quantum,
Then spies of all factions will want 'em.
Our codes will all fail,
And they'll read our email,
Till we've crypto that's quantum, and daunt 'em.

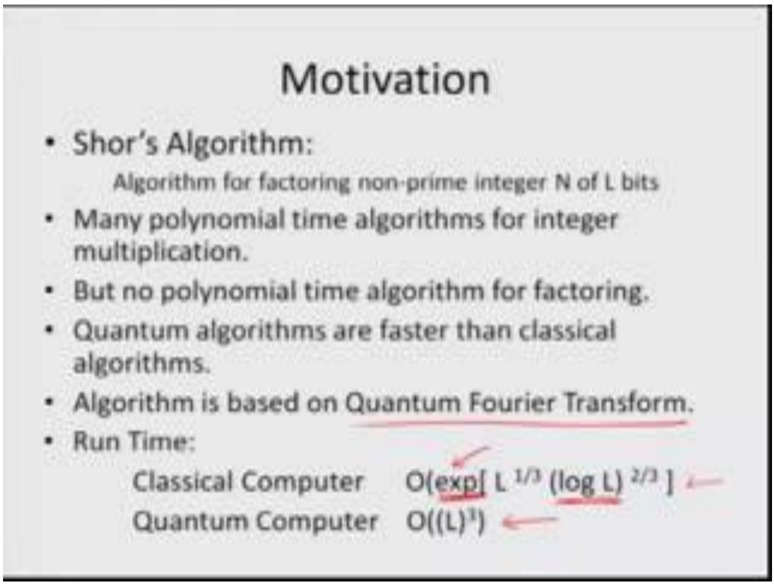
- Invented a **quantum algorithm** for efficiently **finding the prime factors** of large numbers.
- Classical factoring algorithms: $\mathbf{O}((\log N)^k)$
- Shor's algorithm: $\mathbf{O}(\log N)$

.

Here is a code from whom, he said that if computers that you build up a quantum then spies of all factors will want them, our codes will all fail and they will read our email till we have crypto that is also quantum and daunt them all. Basically it is an interesting idea that once you get the quantum advantage; then whoever gets it first has the upper. Because if the spice get it as in mention, they will be having all the benefit and our security goes, but on the other hand once we have the quantum on our side also then we can keep it so secure that once again nobody can wait them.

That is the reason of perusing this particular concept of security to quantum aspects; invented a quantum algorithm for efficiently finding the prime factors of large numbers. The classical factoring algorithms work in the order which is exponential in time where as the Shor's algorithm works in polynomial time. Therefore, it is so impressive and that is exactly the utilizing the full power of the quantum because that is expected when you do a quantum algorithm. The reason being that we all know that quantum sub space is exponential in size compared to the classical substance that is the reason why they should have happened all the time and here is the first case why it was actually shown.

(Refer Slide Time: 03:41)



Motivation

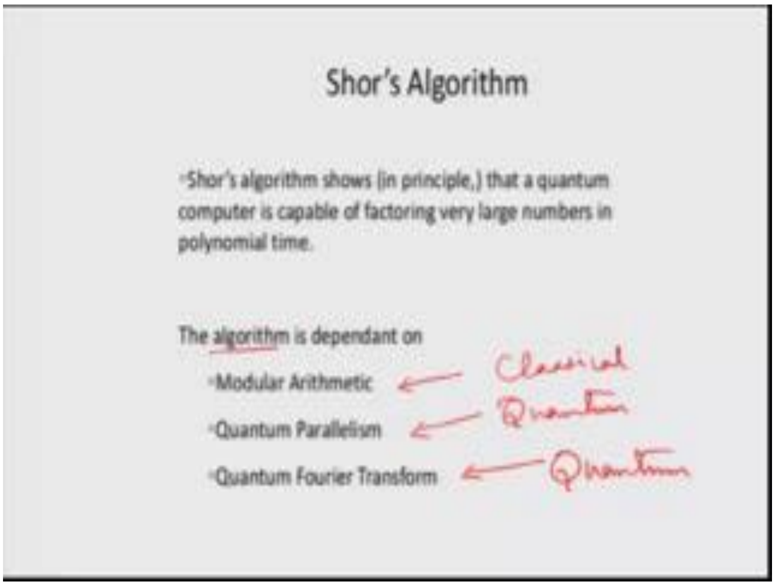
- Shor's Algorithm:
  Algorithm for factoring non-prime integer N of L bits
- Many polynomial time algorithms for integer multiplication.
- But no polynomial time algorithm for factoring.
- Quantum algorithms are faster than classical algorithms.
- Algorithm is based on Quantum Fourier Transform.
- Run Time:
  Classical Computer $O(\exp[\, L^{1/3} (\log L)^{2/3}]$
  Quantum Computer $O((L)^3)$

.

The motivation is plenty, it is for factoring non prime integers of N of L bits many polynomial time algorithms for integer multiplications are available, but there are no polynomial time algorithm for factoring because it is like very easy to find the solution of 2 numbers which are been multiplied which can be primes, but to be able to get the solution or the factorization of a large number in to its primes is a hard problem. Quantum Algorithms are faster than classical algorithms that what will show, algorithm is based on quantum Fourier transform.

This is one of the things which we will also learn as a part of this particular algorithm; where will be learning about quantum Fourier transforms. The run time is the biggest advantage as we have been discussing before for a classical case it is exponential where as in the quantum case it is polynomial. The exponential comes because I have a log sitting here and an exponential sitting here. This is the reason for our exponential.

(Refer Slide Time: 05:06)



Shor's Algorithm

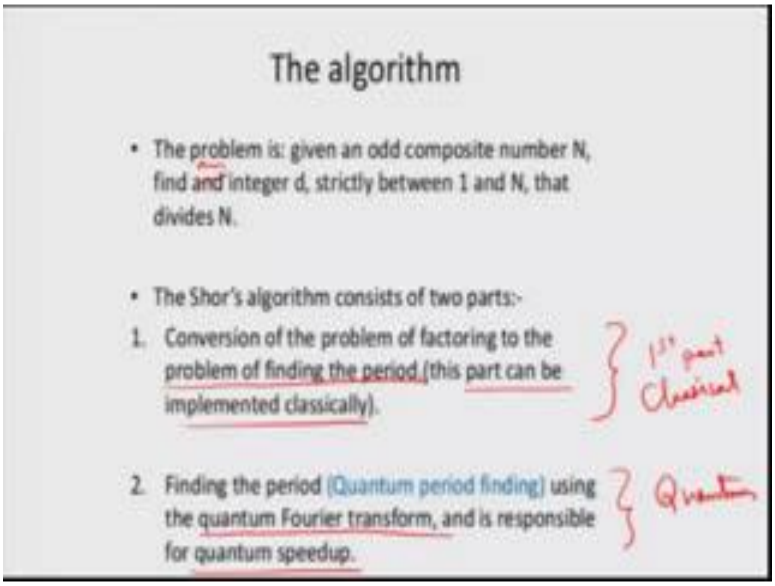
- Shor's algorithm shows (in principle,) that a quantum computer is capable of factoring very large numbers in polynomial time.

The algorithm is dependant on

- Modular Arithmetic ← Classical
- Quantum Parallelism ← Quantum
- Quantum Fourier Transform ← Quantum

The Shor's algorithm shows in principle that a quantum computer is capable of factoring very large numbers in polynomial time. The algorithm is dependent on modular arithmetic principles, quantum parallelism and quantum Fourier transform. The first part which is modular arithmetic is essentially classical is nothing new in it. The quantum parallelism is the part which is quantum and obviously, the quantum Fourier transform is also the part which is quantum.

(Refer Slide Time: 05:51)



The algorithm

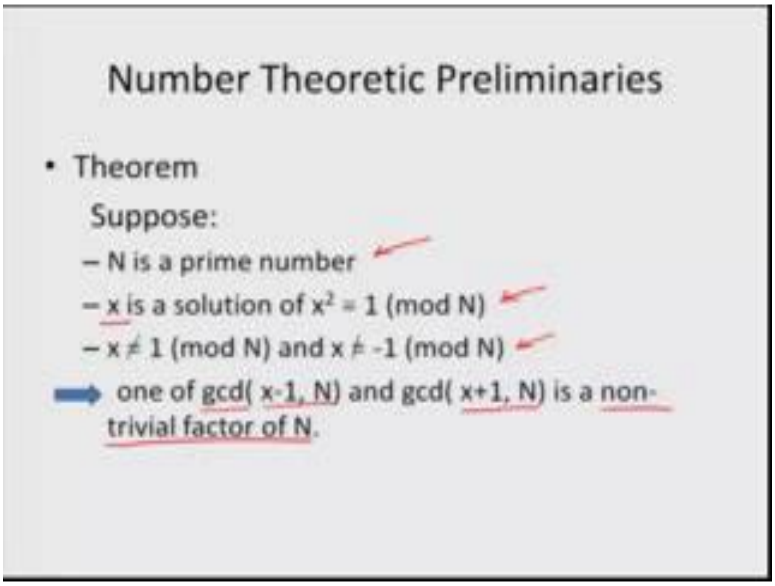
- The problem is: given an odd composite number N, find and integer d, strictly between 1 and N, that divides N.

- The Shor's algorithm consists of two parts:-

1. Conversion of the problem of factoring to the problem of finding the period (this part can be implemented classically). } 1st part Classical
2. Finding the period (Quantum period finding) using the quantum Fourier transform, and is responsible for quantum speedup. } Quantum

Entire algorithm therefore, has essentially parts some of it remains classical as expected. The problem can be stated in such a way, given an odd composite number N find an integer d sorry strictly between 1 and N that divides N. The Shor's algorithm consists of mainly 2 parts. Conversion of the problem of factoring to the problem of finding the period; this part can be implemented classically. This is the part which is the classical as I mention. Second part is the part which is finding the period which involves quantum Fourier transform and this is the one which is responsible for the quantum speedup and it utilizes for the parallelism.

(Refer Slide Time: 06:42)



Number Theoretic Preliminaries

- Theorem

  Suppose:

  – N is a prime number

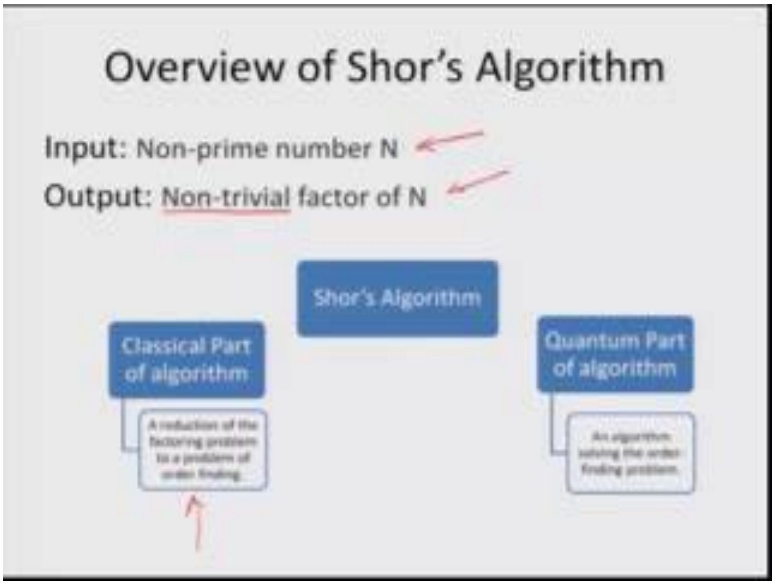
  – x is a solution of $x^2 = 1 \pmod{N}$

  – $x \neq 1 \pmod{N}$ and $x \neq -1 \pmod{N}$

  ⇒ one of gcd(x-1, N) and gcd(x+1, N) is a non-trivial factor of N.

For learning the first part, of addressing the first part which is classical. In order to do the first part which is classical, let us look at some number 30 background. There exist a theorem in the number theoretic principle which says that if N is a prime number and there exist a solution x such that x squared is equal to 1 mod N and x is not equal to mod N and x is not equal to minus 1 mod N. One of the greater common denominators of x minus 1 and N and x plus 1 and N is going to be a non trivial factor of N.

(Refer Slide Time: 07:45)



Overview of Shor's Algorithm

Input: Non-prime number N

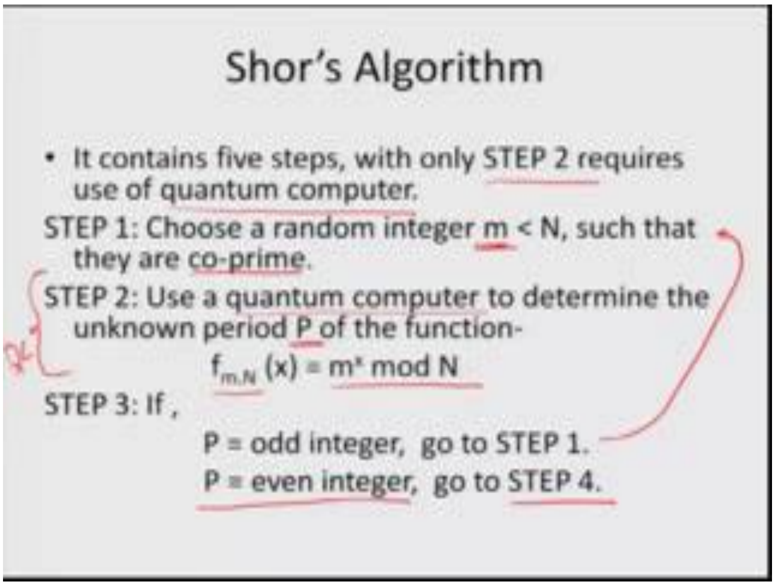
Output: Non-trivial factor of N

Shor's Algorithm
Classical Part of algorithm
A reduction of the factoring problem to a problem of order finding.
Quantum Part of algorithm
An algorithm solving the order finding problem.

.

In terms of what the Shor's algorithm does, is that it inputs the non prime number N and it outputs the non trivial factor of N. The reason why this term non trivial is important is because 1 is always a factor of N. So, we are not really looking for such non trivial factors as 1. In figurative sense, this is how it looks we have the classical part of the algorithm for which we did all this theorem and the quantum part of the algorithm, which is going to do the orders finding problem and the first is going to reduce the factorization problem in to a problem of order finding.

(Refer Slide Time: 08:36)



Shor's Algorithm

- It contains five steps, with only STEP 2 requires use of quantum computer.

STEP 1: Choose a random integer m < N, such that they are co-prime.

STEP 2: Use a quantum computer to determine the unknown period P of the function-

$$f_{m,N}(x) = m^x \bmod N$$

STEP 3: If ,

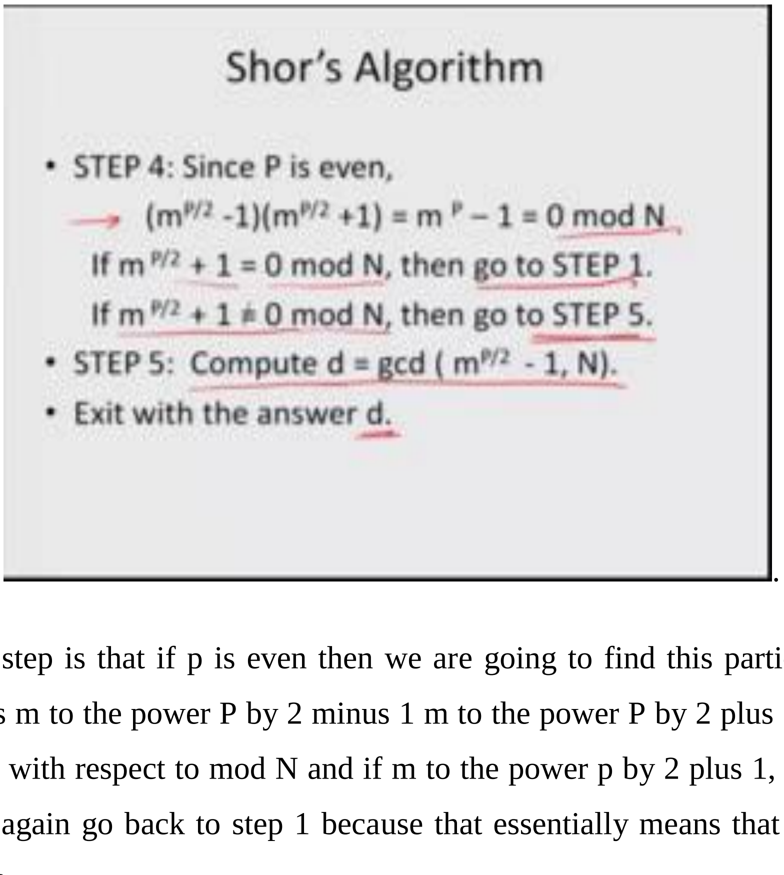
P = odd integer, go to STEP 1.

P = even integer, go to STEP 4.

It consists of 5 steps with only step 2 that requires the use of quantum computer. It is important to note this because if you think that a quantum computer is going to always have every step quantum that is not true. That is the reason why I am also stressing on this part is the algorithms often would be which are even key for quantum computations would still might not need everything to have the machinery of quantum ideas, regular classical concepts would still be very important many cases.

In this case the first step is to choose a random integer m such that it is less than N and they are co-prime. Now the second step is to use a quantum computer to determine the unknown period P of the function f of m N. We have already chosen m and N of x which is equal to m to the power x mod of N. The quantum computer is going to determine the period of this function and finally, in step 3 if the period is an odd integer we go back to step 1, but if the period is an even function then we go to the next step.

(Refer Slide Time: 10:11)



Shor's Algorithm

- STEP 4: Since P is even,

  $(m^{P/2} -1)(m^{P/2} +1) = m^{P} - 1 = 0 \bmod N$

  If $m^{P/2} + 1 = 0 \bmod N$, then go to STEP 1.

  If $m^{P/2} + 1 \neq 0 \bmod N$, then go to STEP 5.
- STEP 5: Compute $d = \gcd(m^{P/2} - 1, N)$.
- Exit with the answer d.

.

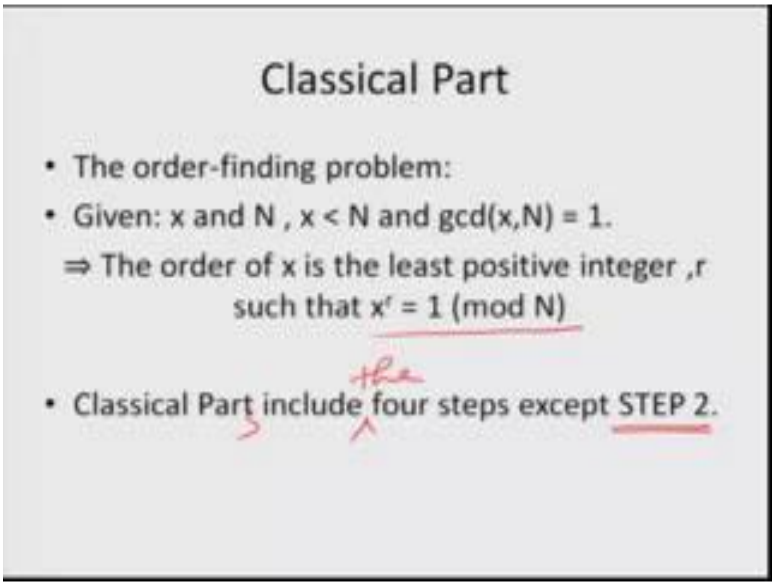
And the next step is that if p is even then we are going to find this particular function form, which is m to the power P by 2 minus 1 m to the power P by 2 plus 1 and find out how it sets up with respect to mod N and if m to the power p by 2 plus 1, 0 mod N then we can go to again go back to step 1 because that essentially means that it is having a trivial solution.

However, if m to the power P divided by 2 plus 1 is not equal to 0 mod N then we go to step 5, which means that we are getting closer to the solution. What we have done until now is basically shown you the most general way of looking at which can be sometimes boring we will go to the specific examples soon to make it much more easier, but as if now let us just complete the difference steps. Finally, when we reach step 5 we are going to compute the greatest common divisor GCD for this particular step that we just found out from earlier step and the answer of that is going to be our solution.

In operational principles there are 5 steps, but you can see only step 2 (Refer Time: 11:50) quantum step. The first step once again was the choosing of a random integer which did not require any quantum computer, except that when we choose we have make sure that there co-prime. Second was to make sure that we get the quantum computation part 2, go properly to find the unknown period of the function. This is the only QC part, first step non QC, third step again once we have found that is to continue by principles which are again classical computation wise.

For instance we just noted whether our solution was one kind of the other and then we computed whether the mod values of them in a certain form of the other could let us us go back to the first step essentially showing indicating trivial solution or go in to the forward steps were we finally, computed get its common deviser to exit with the right answer. So, that is the basic principle of the Shor's algorithm.

(Refer Slide Time: 13:10)



Classical Part

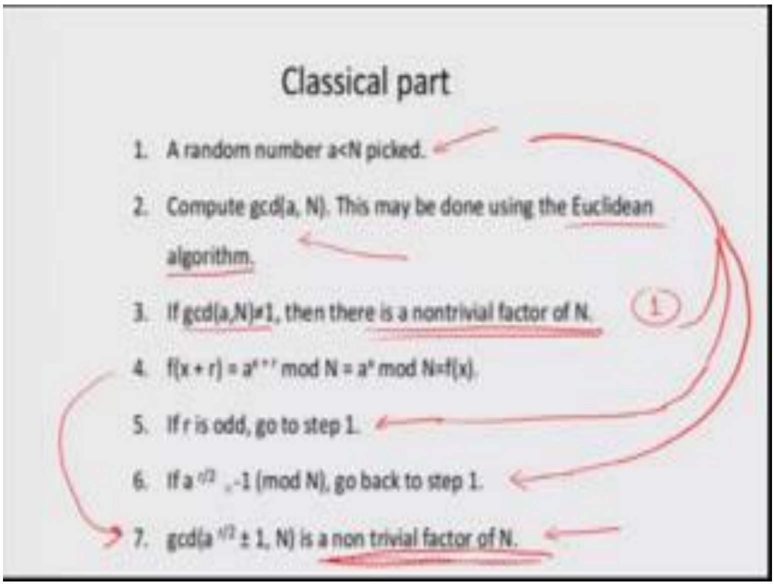
- The order-finding problem:
- Given: x and N , x < N and gcd(x,N) = 1.

⇒ The order of x is the least positive integer ,r such that $x^r = 1 \pmod N$

- Classical Parts include the four steps except STEP 2.

Now, here is the classical part which is the order finding algorithm, in this given x and N where x is less than N and there is common deviser of the 2 is 1, Ensures that you are getting co-prime numbers right. The order of x is that the least positive integer r such that x to the power r is equal to 1 mod N so, classical parts include the 4 steps except step 2 as I mentioned that.

(Refer Slide Time: 13:45)



Classical part

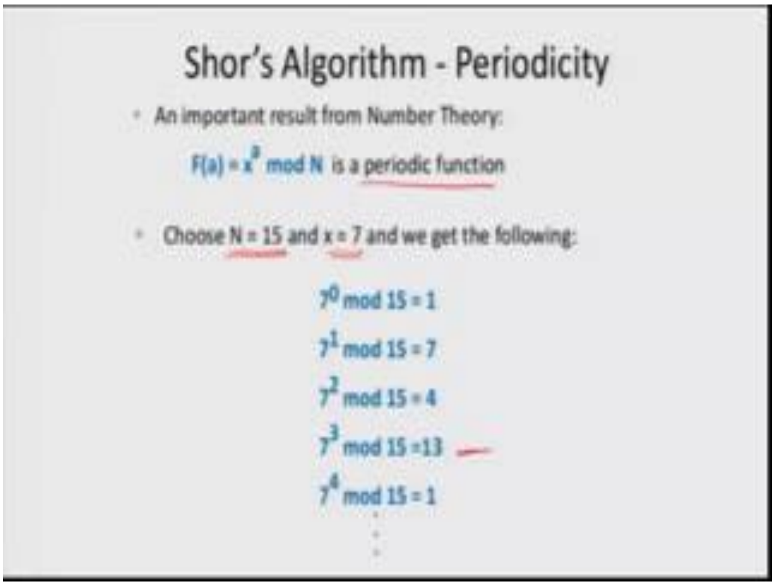
1. A random number a<N picked.
2. Compute gcd(a, N). This may be done using the Euclidean algorithm.
3. If gcd(a,N)≠1, then there is a nontrivial factor of N.
4. $f(x + r) = a^{x+r} \bmod N = a^{x} \bmod N = f(x)$.
5. If r is odd, go to step 1.
6. If $a^{r/2}$,-1 (mod N), go back to step 1.
7. gcd($a^{r/2}$ ± 1, N) is a non trivial factor of N.

.

The classical part consist of a random number a which is less than N which is picked; we compute a of N, this may be done using the Euclidean algorithms. Then we find the gcd and if it is 1, if it is not equal one then it is a non trivial factor which is essentially as we said 1; we do not use that you go back otherwise we go forward compute this to find the value of r and if r is 1. Once again go back to 1 because that is not going to make a square in a further; however, if it is not that then we continue and we get if r is odd we get back to step 1 because we have to read with the problem again if it is minus 1 mode of 1 again we go back to step 1.

These to all of these 3 goes back to step 1, Only 4 is the one where we can do and get to the point where we can get a 7th step we can get the non trivial factor of N. That is these are the important classical steps that we have to do in this process.

(Refer Slide Time: 15:19)



Shor's Algorithm - Periodicity

- An important result from Number Theory:

  $F(a) = x^a \bmod N$ is a periodic function

- Choose N = 15 and x = 7 and we get the following:

  $7^0 \bmod 15 = 1$

  $7^1 \bmod 15 = 7$

  $7^2 \bmod 15 = 4$

  $7^3 \bmod 15 = 13$

  $7^4 \bmod 15 = 1$

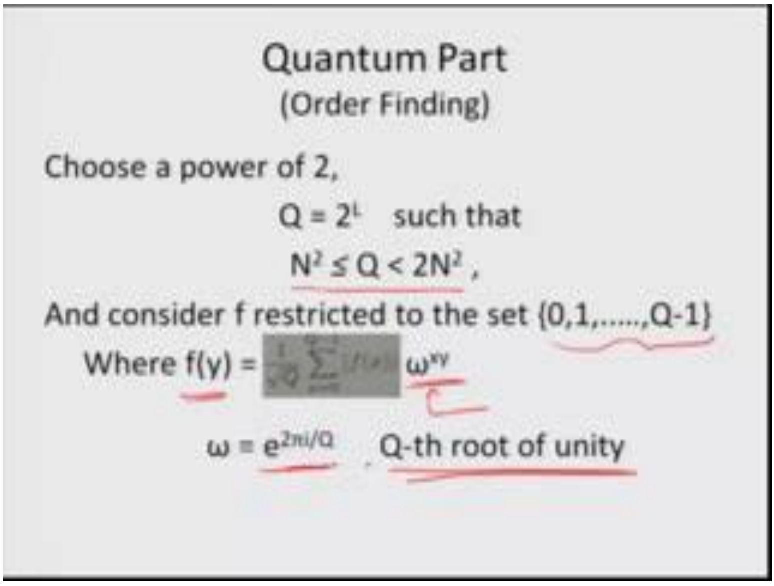
  $\vdots$

.

First we are using the important step, important result from the number theory where we are converting into periodic function problem, here is an example; that we first check. Let us take the case where my number is 15, although this is a trivial problem, but let us see if you can understand the problem. So, I am going to choose 15 which we are going to do factorization. Let us choose the number, that we said has to be co-prime with respect to the original number.

If you take 7, x is equal to 7, what do we get? So, x to the power r mod of that number as we discussed are the numbers that we are looking for. So, r can be any number so we have going through this numbers, r can be any number and less than 7. 7 to the power 0 mode 15, 1 into power 1 mod 15 so on and so forth. We see that it starts repeating after the number 3. 0, 1, 2, 3 and its start repeating 4 is again one again equals to the circle.

(Refer Slide Time: 16:36)



Quantum Part

(Order Finding)

Choose a power of 2,

$Q = 2^L$ such that

$N^2 \leq Q < 2N^2$,

And consider f restricted to the set {0,1,.....,Q-1}

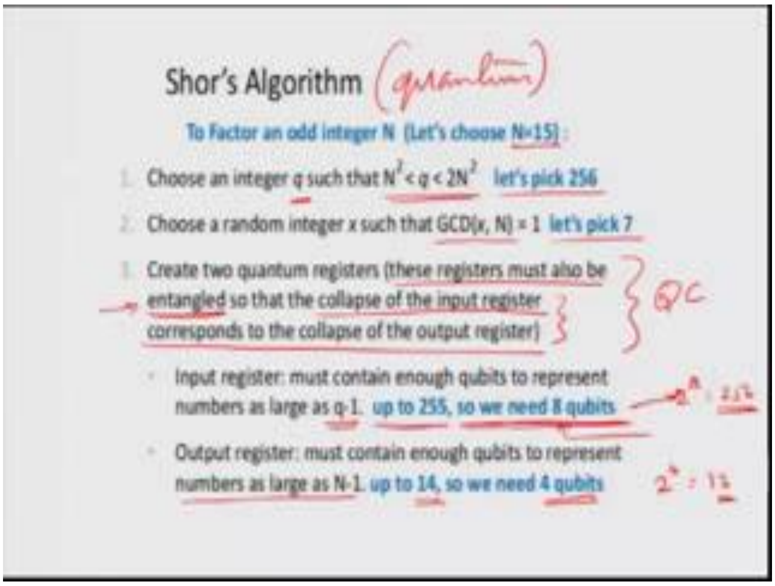
Where $f(y) = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} f(x)\, \omega^{xy}$

$\omega = e^{2\pi i/Q}$, Q-th root of unity

This we could look at in a classical way for a small trivial number, but what we are trying to look for is to figure out how this order finding can be done when you do it can I quantum way. Let us choose a power of 2, let us say Q is equal to 2 to the power l such that this Q lies in between N square and 2 N square and if we consider f is restricted to the set. Our set that we are looking for restricted to the set where it goes between 0 and Q minus 1 such that our function is given by this normalized value with respect to omega x y where omega is the part which will give us the period. So, exponential 2 to the power pi I over Q, is the way how you look at the period and that is our Qth root of unity.

This is a mod generalize way of looking into the problem as to how we are going to the order finding. In the classical way we just looked at it last time we can do it step by step and we can see how it starts repeating. Now that is possible when numbers are small we are looking at simple solutions; however, when we look at harder (Refer Time: 18:09) more difficult cases, we can actually formulate a process and this is the how you can actually formulate; you can use the phase of the problem in terms quantum system to get your solution that is the idea here.

(Refer Slide Time: 18:22)



Shor's Algorithm (quantum)

To Factor an odd integer N (Let's choose N=15):

1. Choose an integer q such that $N^2 < q < 2N^2$ let's pick 256
2. Choose a random integer x such that GCD(x, N) = 1 let's pick 7
3. Create two quantum registers (these registers must also be entangled so that the collapse of the input register corresponds to the collapse of the output register) QC
   - Input register: must contain enough qubits to represent numbers as large as q-1. up to 255, so we need 8 qubits $2^8 = 256$
   - Output register: must contain enough qubits to represent numbers as large as N-1. up to 14, so we need 4 qubits $2^4 = 16$

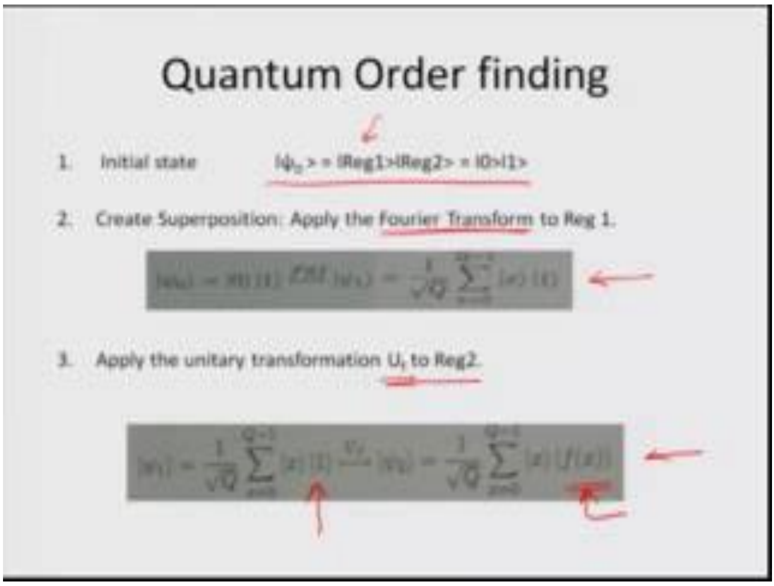
In what Shor's was able to recognize is that in order to factor an odd integer N let us choose for example, 15 you have to go through the steps in this particular way which we discussed here the algorithm takes this form which is perhaps able to understand.

So, we have choose an integer Q such that now Q, such that it lies in between N square and 2 N square so we said. It can be any number let us pick up 256 because its 2 to power something. Choose a random number x such that the gcd of x N is 1. We can pick a number like 7 so you did earlier. Create 2 quantum registers, these registers must also be entangled; now we have discussed about entanglement before and that is the part which is critical in Shor's algorithm will again revisit that slightly more detail later on, but for now just key look up your notes or previous lectures go back to understanding what entanglement is. So, these registers have to be entangle.

That the collapse of the input register corresponds to the collapse of the output register right. I mean this is the criticality of entanglement that one process essentially and ensures the other one is also correlated one way of the other. The input register therefore, must contain enough qubits to represent number as large as Q minus 1 that is up to 255 in this particular case. So, we need 8 qubits, 2 to the power 3 qubits which can give us up to 255.

We need the input register to contain enough qubits to represent numbers as large as Q to the power large as Q minus 1 which is up to 255 so, that we need 8 qubits since 2 to the power 8 is 256 therefore, we choose the number of qubits is 8. The output register again must contain enough qubits to represents numbers is larger than minus 1 up to 14, so basically need 4 qubits, 2 to the power 4 is good enough to go up to 120; let us more than what we need.

(Refer Slide Time: 21:01)



Quantum Order finding

1. Initial state $|\psi_0\rangle = |Reg1\rangle|Reg2\rangle = |0\rangle|1\rangle$
2. Create Superposition: Apply the Fourier Transform to Reg 1.

$$|\psi_0\rangle = |0\rangle|1\rangle \xrightarrow{QFT} |\psi_1\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle|1\rangle$$

3. Apply the unitary transformation U_f to Reg2.

$$|\psi_1\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle|1\rangle \xrightarrow{U_f} |\psi_2\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle|f(x)\rangle$$

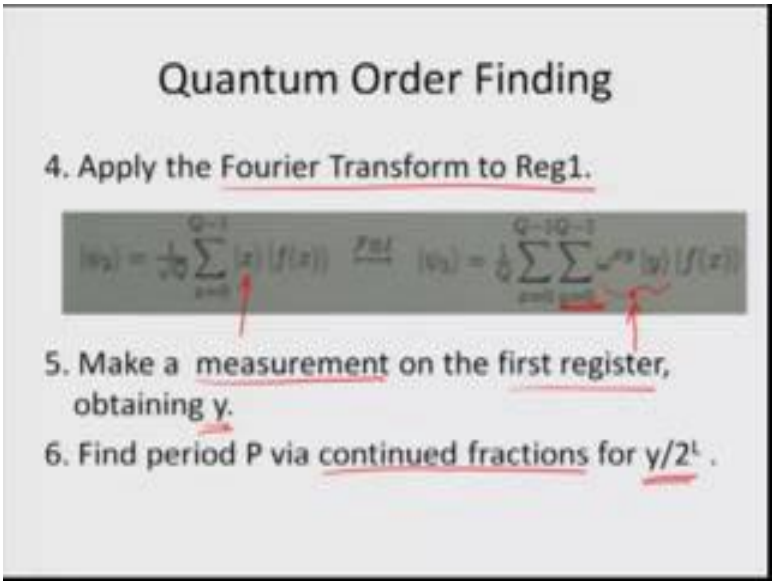
.

Now, in order to look at the problem of order finding, quantum order finding let us now go back and look at how these states can be set up.

Let us setup the states 0 and 1, register 1 and register 2 which is our initial state we create a super position; that is we apply the Fourier transform to register 1 now. This principle of Fourier transform is something which is perhaps known to you from previous studies, but if not we have a separate lecture were we will be focusing on Fourier transform and the quantum Fourier transform principles. However, it comes under the basics of the mathematical formulation that we have been mentioning before. So, we can look back on to that once again if necessarily.

Once you apply the Fourier transform argument, then what you will have is will have this entire state; which you look like this and then you can apply the unitary transformation to the register 2, which is your wish to register 2 which takes it to this functional f starting instead of the 1 function. This is the unitary function which puts in the f of x there.

(Refer Slide Time: 22:37)



Quantum Order Finding

4. Apply the Fourier Transform to Reg1.

$$|\psi_3\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle|f(x)\rangle \xrightarrow{FT} |\psi_4\rangle = \frac{1}{Q}\sum_{x=0}^{Q-1}\sum_{y=0}^{Q-1}\omega^{xy}|y\rangle|f(x)\rangle$$

5. Make a measurement on the first register, obtaining y.
6. Find period P via continued fractions for $y/2^L$.

.

Once we get this form then we can actually apply the Fourier transform again to register 1 which is the x and we are going to get this as our Fourier transformed argument as a result of application of the Fourier transform. We make a measurement on the first register now which gives us the value of y because that is the part which we have now continent to we find the period P by using the continued fraction method for y over 2 to power L.

These are the 6 steps, which we have just discussed for your quantum part of the algorithm. So, let me actually reiterate once before we close this lecture. What did we do? We basically said that we take up a trivial case. Were we apply from the number theory principle the periodic function principle and we found out that it was possible to find out a value of 7, which is co-prime with 15 which is the number we have trying to find out the factors for. And for this particular kind of a case where we can do each and every number in terms of finding x to the power a mod N; we can find the periodicity by hand because this is easy, but this gives you the idea as to how will be setting this up this problem up in the quantum form.

So, in the quantum form what we did was we choose the numbers and the registers in such a way, so that we can set up the order finding problem. We choose in terms of powers of 2 because that is how the quantum is suppose to be we said these functions in the sets which had the registers in those connections and we applied our functions such that we got the period function coming up and that was possible because of the application of Fourier arguments, where the phase of the problem comes in to the picture and we got the Qth root of unity for instance the omega value here and we went through the individual steps in particular when we did this particular example of N equal to 15 the quantum way.

Once again even in this quantum way we had steps which did not need quantum concept; however, because it needed entanglement, classical concepts here mathematically the movement we have actually use qubits we cannot really start measuring at every steps. It remains quantum in that sense and so we have to choose our number of registers, the number of qubits in a very important way and that is what we have done here. And we have shown how this particularly it will look like in its final step of which register to use and which part to apply the transforms to and what are the final results that will be getting when we apply our Fourier transforms to the final result. So, that we can finally, find the period by using the continued fraction (Refer Time: 26:18).

With this I am going to end the class today, is you have gone through a lot of concepts and we are going to revisit parts of it to make sure that you understand it more as we go ahead with this particular approach of algorithms.

Thank you.