Introduction to Cryptology
Dr. Sugata Gangopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee

Lecture – 13
Primality Testing: Miller-Rabin Algorithm,
Legendre Symbol and Jacobi Symbol

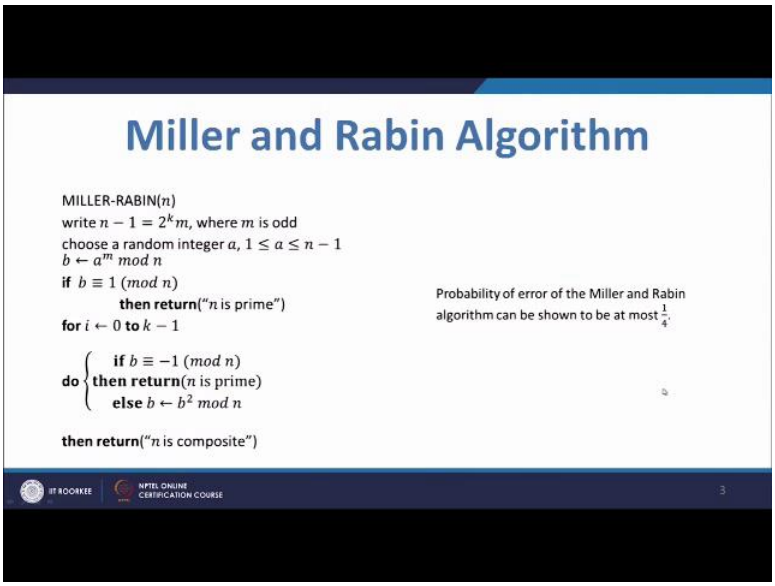
Welcome to week three lecture three. In this lecture, we will discuss primality testing. Now, in our discussions of RSA algorithm, we found that, one of the most important things in setting up an RSA cryptosystem is that, we have to have large primes. And, these primes have to be about 600 bits long.

The question that occurs to us that, how are we going to find out such large primes; and, that is what we are going to discuss today. So, we will be discussing algorithms, which decide whether a number is prime or not. And, these algorithms are called primality testing algorithms. One point of question here is that, these algorithms are not deterministic algorithms. So, the results of these algorithms are not correct always. What happens essentially is that, suppose we have a primality testing algorithm, which is not deterministic, we will be discussing this algorithm soon.

These algorithms are I mean if we take a number – a positive odd integer, we apply the algorithm on it; then, the algorithm will return the input is prime or the input is composite. These algorithm are constructed in such a way if n is prime, it will determine n is prime without error. But, there are cases, where it may decide that, n is prime although the n is, although the concept, the input is not prime. In that case, it will have error.

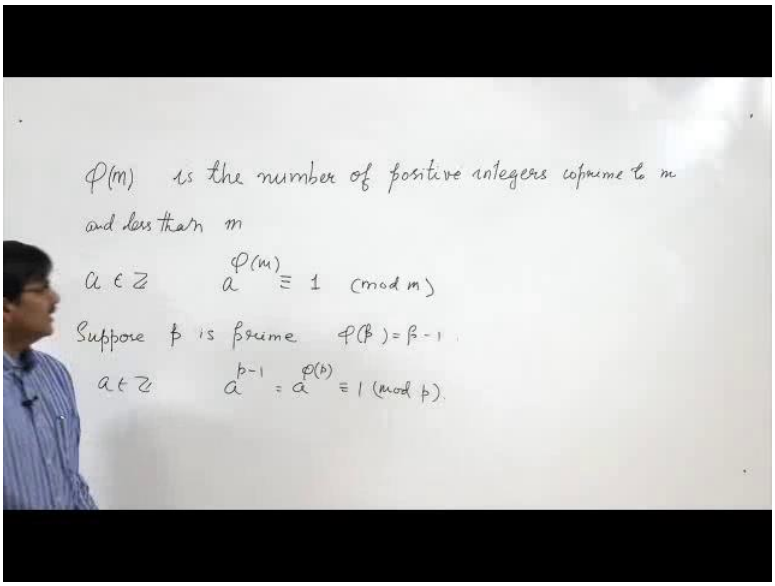
Then, after designing such an algorithm, people compute the error probabilities of these algorithms. And, they found that, the error probabilities are bounded; that is, to say that, if they found that the error probabilities are bounded and the bound is reasonable, then we know that, we can work with these algorithms. We will look at these things a little later. But, first we look at our first algorithm.

(Refer Slide Time: 03:09)



This is called Miller and Rabin algorithm.

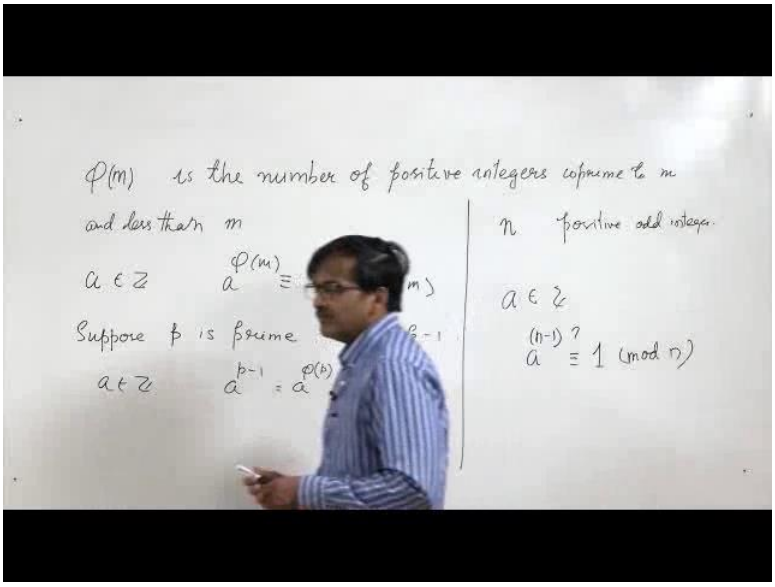
(Refer Slide Time: 03:22)



Now, before going into Miller and Rabin algorithm, we will discuss something that we have already done before that, we have defined phi m; where, m is a positive integer and we have said that, phi m is the number of positive integers coprime to m and less than m. Now, we also know that, if we take any integer a and then if we take a raised to the power phi m; then, this is congruent to 1 mod m. This is something that we have proved in our previous lecture.

Now, suppose m is prime; then, phi m is equal to m minus 1. This is because that, any integer less than m and of course greater than or equal to 1, is coprime to m since m is prime. So, let us change m to p. So, here we write this. So, it looks more like prime. But, if we apply this result here; then, we know that, if I take any element a in z; then, a raise to the power p minus 1, which is equal to a raised to the power phi p is going to be congruent to 1 mod p. Now, we question that, what happens if we run it the other way round.

(Refer Slide Time: 05:59)



Now, suppose we take a number p; number n let us say n; we want to determine whether n is prime or not. First of all, if n is even; then, there is no point checking it, because we know that, even number cannot be a prime. So, n has to be positive odd. And now, what we do is choose an a, which is inside z and then take a raised to the power n minus 1; and, ask whether this is equal to 1 mod n. Now, it may or may not be 1 mod n if n is prime; of course, it is going to be 1 mod n; but, otherwise also, it may be 1 mod n. What we do in this algorithm, if it is 1 mod n; we say that, it is a prime; but, if it is not 1 mod n, we know for definite that n is not prime. And, this is the strategy.

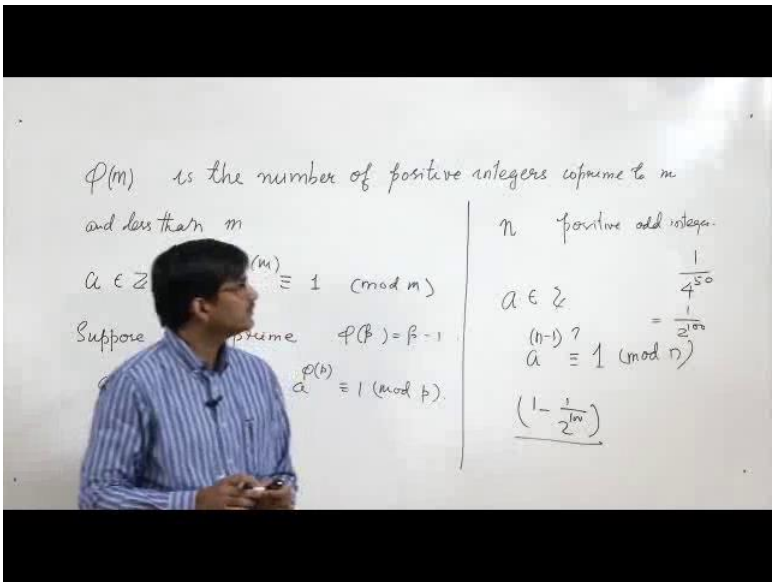
And now, let us look at the slide. Here it is Miller and Rabin algorithm. In this algorithm, we have made some small changes to make the algorithm work better. So, I have the input n; and of course, it is odd. And now, after that, I am reducing n by 1 and then I am taking the factor of 2. So, I know that, if n minus 1 is an odd – is an even integer; so, I

factor it by 2 to the power k for maximum k. So, it gets split up into 2 to the power k into m; where, m is odd. Now, I choose at random a between 1 and n minus 1.

And, after that, I raise a to the power m mod n and store it in b. Now, I check whether b is 1 mod n. If it is 1 mod n, I say that, n is prime. Well, I may be wrong, but I say that, m is prime; otherwise, I go into a loop; I basically keep on multiplying 2 with m one after another up to 2 to the power k; or, in this case, 2 to the power k minus 1 and check whether a raised to the power 2 to the power i into m is equal to 1 mod – is equal to minus 1 mod n. That is what we do in this step. If you look at this step carefully, you will find that, that is exactly what is happening.

If we find that, at any iteration of this loop b is congruent to minus 1 mod n, we say that, n is prime; otherwise, we return n is composite. If we look carefully, we will see that, we are essentially using a kind of idea that we discussed here in a little refine manner and it will give me; well, an idea whether n is prime or not. Now, what is interesting here is that, this algorithm has a very nice error bound of error probability bound; and, that bound is one-forth. We know that, probability of error of the Miller and Rabin algorithm is utmost 1 by 4.

(Refer Slide Time: 10:26)

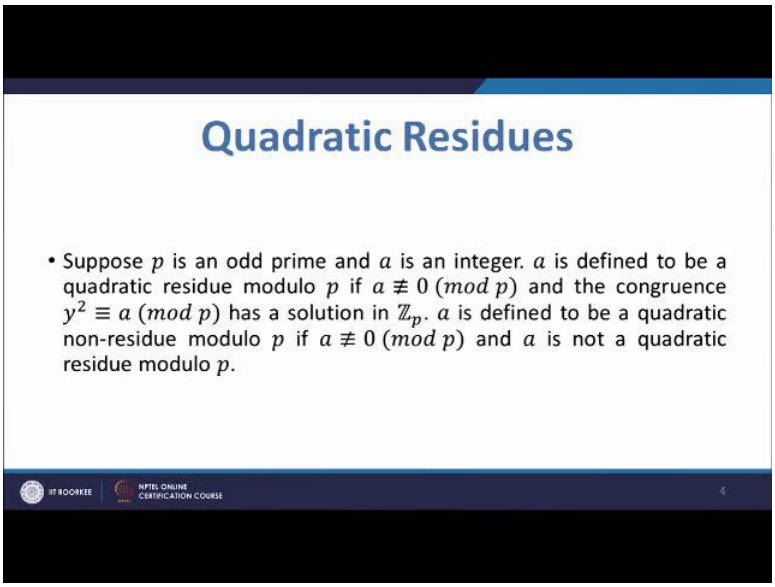


So, therefore, if we run Miller and Rabin algorithm, let us say 50 times; then, the probability that, each time this algorithm is giving me wrong answer is 1 by 4 to the power 50, which is equal to 1 by 2 to the power 100. So, the probability that it is giving

me correct answer is 1 minus 1 by 2 to the power 100, which is very large. So, this is an algorithm that can be used in practice to determine whether a large num – positive odd integer is prime or not.

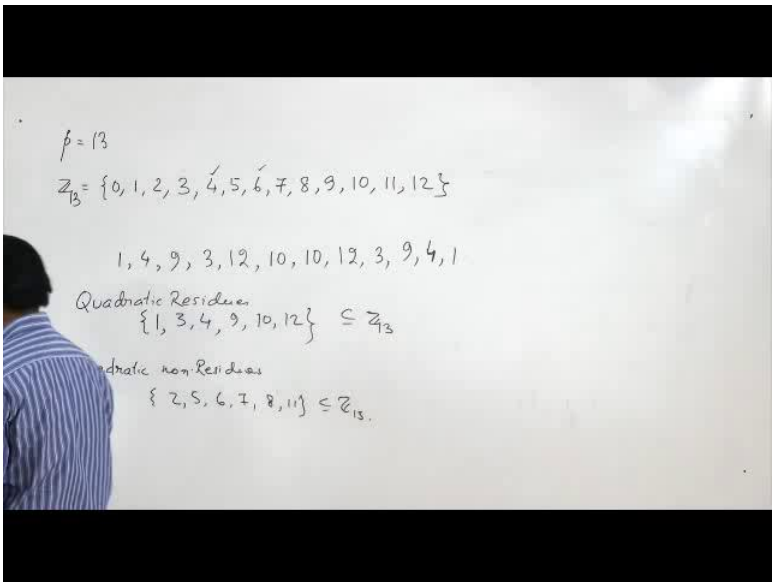
There is another algorithm, which does the same thing and which is also commonly used. And, that algorithm is called Solovay-Strassen algorithm. But, before we go into that algorithm, we have to learn a little more number theory. We come to a topic in number theory, which is called quadratic residues. So, we will discuss that. Suppose p is an odd integer;

(Refer Slide Time: 11:41)



Suppose p is an odd prime and a is an integer, a is defined to be a quadratic residue modulo p if a is not congruent to 0 modulo p and the congruence equation y square congruent to a modulo p has a solution in Z sub p. a is defined to be a quadratic non-residue mod p if a is not congruent to 0 mod p and a is not a quadratic residue modulo p. Now, let us look at some example to check, which numbers are quadratic residue mod p and which are not by fixing a p. We fix p to be 13.

(Refer Slide Time: 12:44)



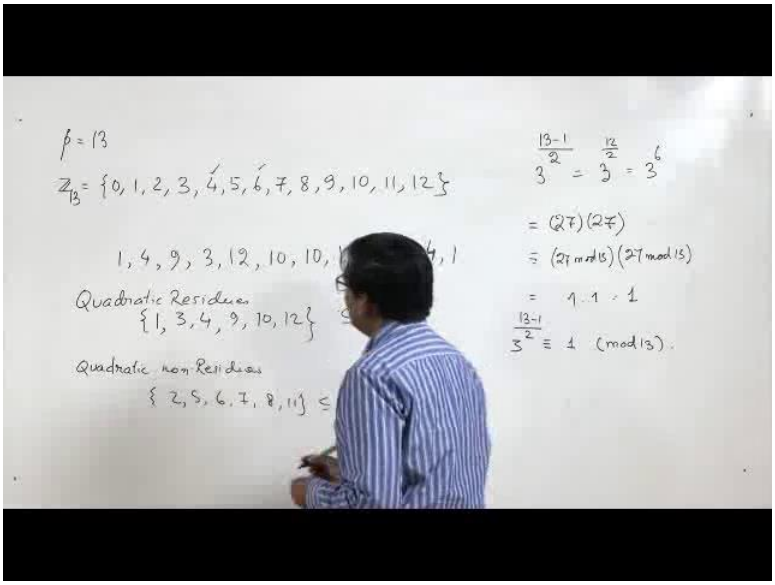
So, we have p equal to 13; of course, it is an odd prime. And, we consider z 13, which are numbers from 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. So, we have here z 13. And then, what we do is that, we take the squares of all the number and we leave aside 0, because we are not concerned with 0. So, if we take the squares and take modulo 13, then we will get a set like this; or, I should rather say a sequence, because we will have repetitions.

So, we will have a sequence like this – 1, 4, because 1 square is 1, 2 square is 4, 3 square is 9; then, we have got 16. But, 16 modulo 13 gives me 3. So, I have come up to 4; 5 square is 25; and, 25 modulo 13 gives me 12. 6 square is 36; 36 modulo 13 gives me 10, because 13 2's are 26; then, 7 also will give me 10; and then, 8 square mod 13 will give me 12; 9 square mod 13 will give me 3. And similarly, I will have for others – 9, 4; and eventually, 12 square mod 13 will give me 1. So, I have got the set of quadratic residues, which are 1, 3, 4, 9, 10 and 12. So, inside z sub 13, these are quadratic residues and the rest are quadratic non-residues apart from 0; we do not say anything about 0. Rest namely, 2, 5, 6, 7, 8, 11. These are also elements of z 13.

Now, we have a famous criterion; it is called Euler's criterion, which says that, in order to determine whether a number is quadratic residue or not, take it to the power p minus 1 divided by 2 and take module of p, that is, reduce it modulo p; if it 1, then it is a

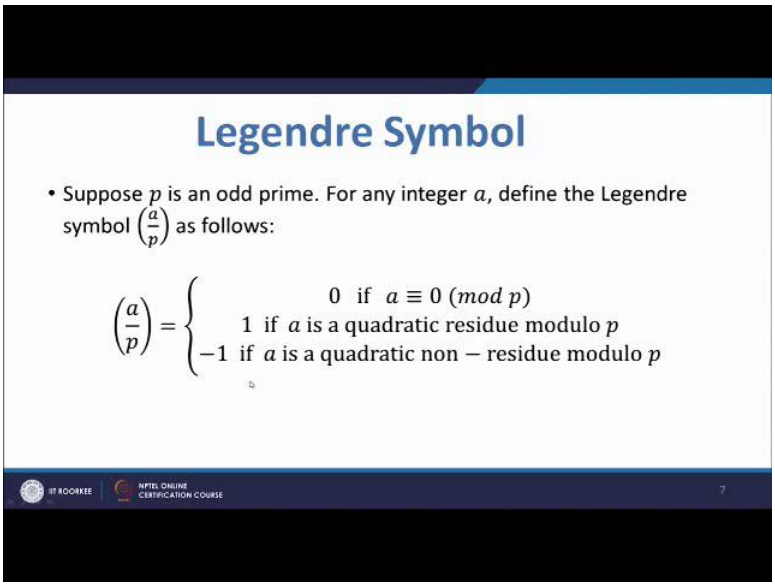
quadratic residue. If it is not 1, then it is not a quadratic residue – then it is not a quadratic residue. So, let us look at that.

(Refer Slide Time: 16:17)



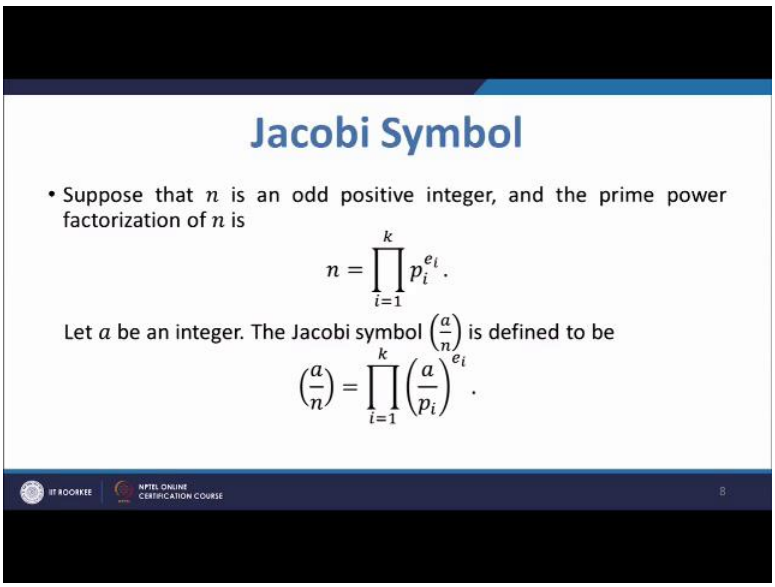
For example, if I start with 3; my prime is p; I take 3. So, 13 minus 1 divided by 2, which is equal to 13; and, this is 12 by 2. This is equal to 6 – 3 raise to the power 6. So, this is 27 into 27. And, if I reduce modulo 13; this one, this is congruent to 27 mod 13 into 27 mod 13, which is 1 into 1. So, it is 1. So, I can say that, 3 raised to the power 13 minus 1 by 2 is congruent to 1 mod 13. And therefore, by Euler's criterion, this is a quadratic residue. If you check another number from the other side, a quadratic non-residue and do the same thing; then, you will see that, you are getting minus 1 mod 13 and not 1 mod 13; and, it is not a quadratic residue.

(Refer Slide Time: 18:03)



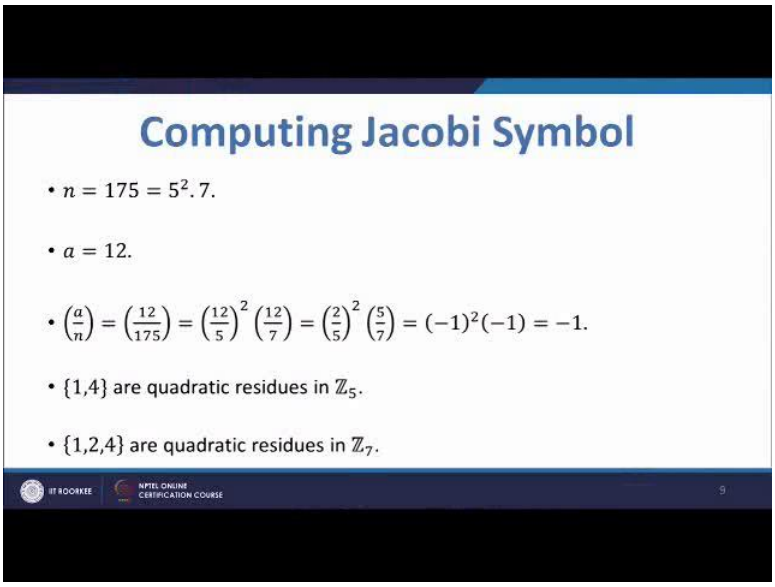
So, ultimately, we will land up into a very famous symbol, which is called Legendre Symbol. If we have an odd prime p and take any integers a; a by p enclosed in parenthesis is set to be Legendre symbol and, it is 0 if a is congruent to 0 mod p. It is 1, if a is a quadratic residue mod p; it is minus 1, if it is a quadratic non-residue mod p. Now, we have a generalization of Legendre's symbol and it is called Jacobi symbol. And, Jacobi symbol can be used for n's, which are not primes or odd primes. But, we take any odd integer n. So, let us see how we define Jacobi symbol.

(Refer Slide Time: 19:08)



So, if n is an odd positive integer and the prime power factorization of n is n is equal to product of i going from 1 to n – p i raised to the power e i; then, the Jacobi symbol a by n is defined to be a by n, is equal to product of the individual Legendre symbols of corresponding to p i's with respect to a raised to the power e i, which is written over here, is a Jacobi symbol. So, if we would like to calculate Jacobi symbol for a particular pair, let us come to the next slide.

(Refer Slide Time: 20:12)

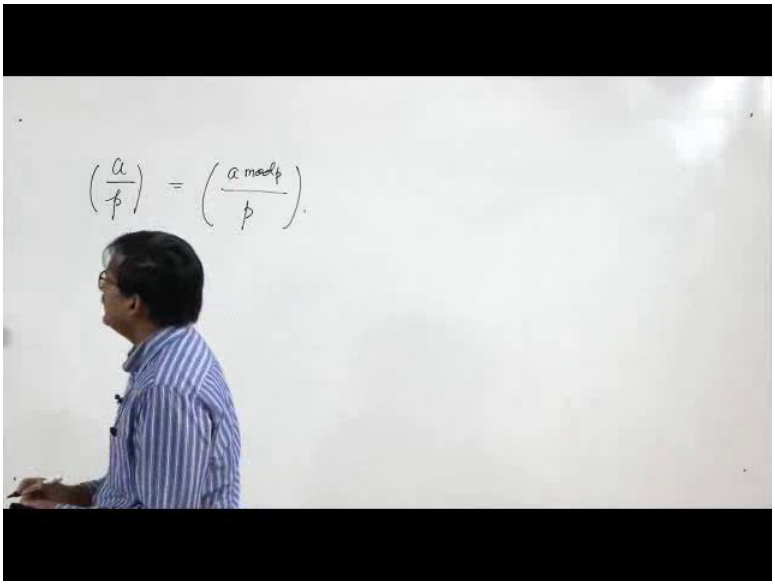


Here we have n, which is 175. If it is an odd, positive integer; and, it factorizes as 5 square into 3 – 5 square into 7. So, that gives me 175. And, let us take a equal to 12. We would like to calculate the Jacobi symbol of a by n. So, it is 12 by 175. And then, according to our definition; let us look at the definition quickly that, we have factorized this and then the Jacobi symbol is product of the Legendre symbols. So, we do this thing. So, we know that, we have got 5 square. So, we write 12 by 5 whole square and then 12 by 7.

Now, it is not difficult to see that, the Legendre symbol value with respect to, or modulo p of m will be same as a Legendre symbol value of m – reduce mod p. So, what we do over here is that, we take 12 and we have 5. We have to calculate the Legendre symbol with respect to modulo 5. So, what I wanted to say is that, if we reduce 12 modulo 5, we get 7. The Legendre symbol value of 12 mod 5 and 7 mod 5 – both are going to be same. No, no, no; I am making a mistake here, if I reduce 12 mod 5, that is, I will get 2. What I

want to say is that, the Legendre symbol value of 2 mod 5 – 2 modulo 5 and 12 modulo 5 are same. This is not difficult to see.

(Refer Slide Time: 22:11)



So, suppose I want to calculate the Legendre symbol value of this; it will be same as a mod p by p. That is what we are using over here. So, I reduce it to 2 5 whole square and 5 7. And now, we can directly calculate the Legendre symbols by using the Euler's criterion. If we do that, we will see both are minus 1. And then, if we calculate the product; then, I will get minus 1. So, we have got the value of Jacobi symbol.

Now, the problem of this method of computation is that, it requires factorization of n; and, which is going to be difficult if n is a large number. In the next lecture, we will discuss how to calculate Jacobi symbol when n is a large number without factorization by using certain properties of Jacobi symbol. And, that will lead us to the Solovay-Strassen algorithm for primality testing.

So, this is the end of Lecture - 3.