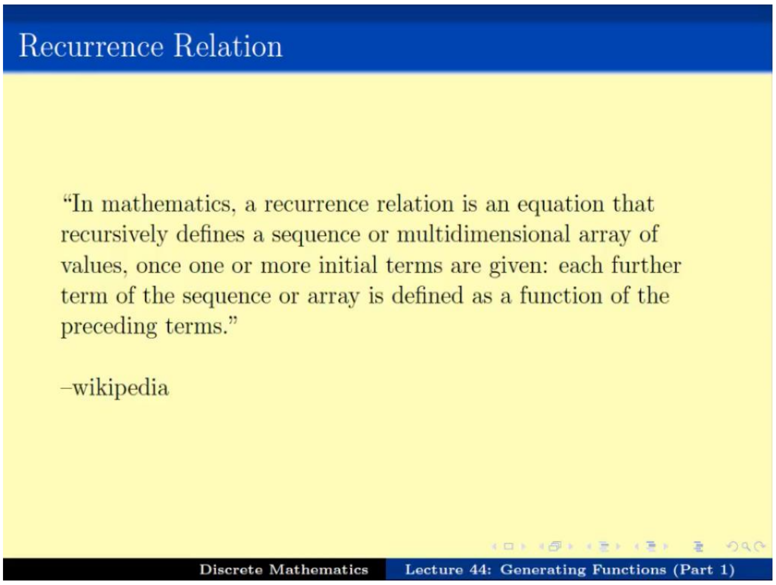
Discrete Mathematics
Prof. Sourav Chakraborty
Department of Mathematics
Indian Institute of Technology – Madras

Lecture - 44
Generating Functions (Part 1)

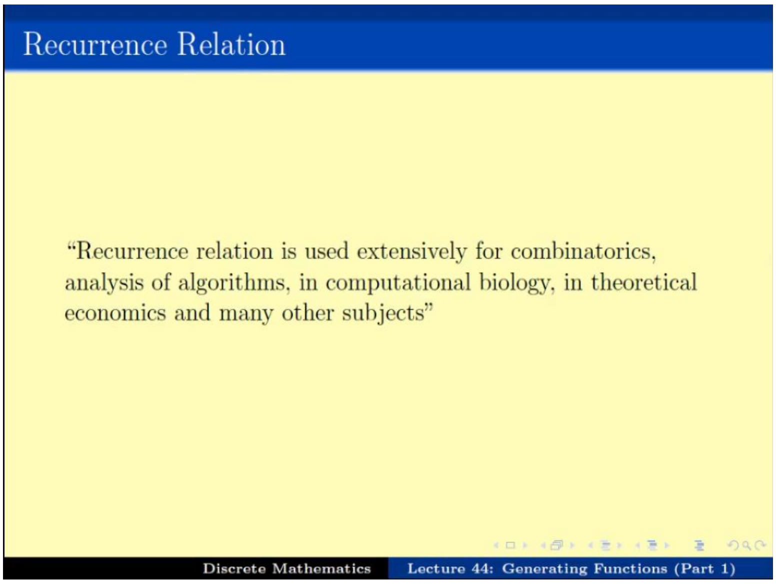
Welcome back. So we have been looking at recurrence relations and how to solve them. In this next set of video lectures, we will see one more technique of solving the recurrence relation and in fact this is one of the most powerful techniques that is there for solving recurrence relations.

(Refer Slide Time: 00:25)



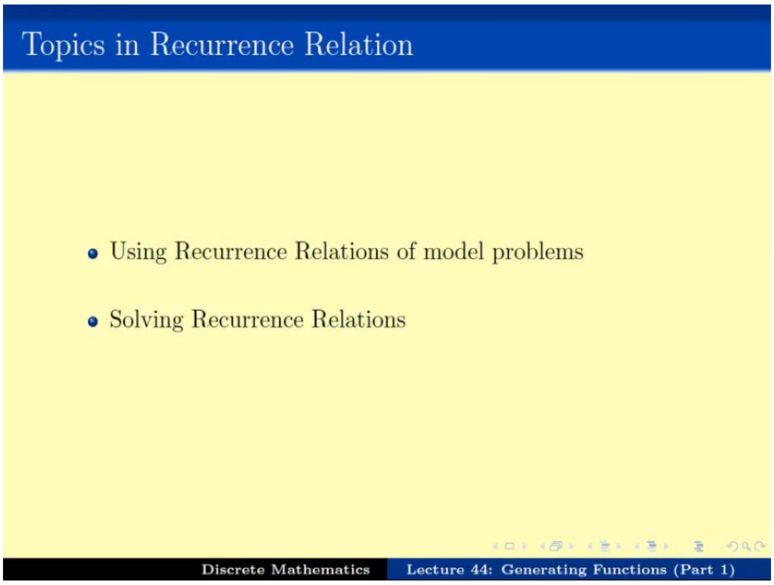
So to start with let us recap. The recurrence we mean a sequence of numbers but the initial set of them are given while nth term is written as a function of the previous terms.

(Refer Slide Time: 00:41)



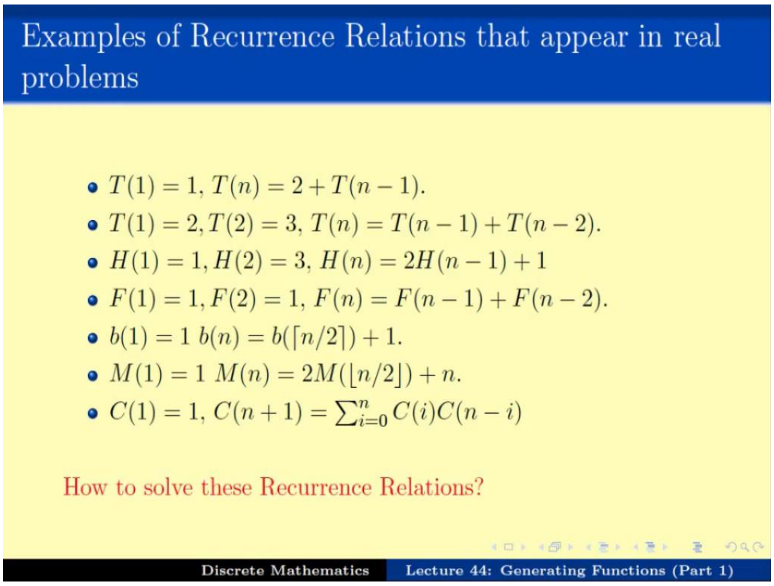
Recurrence relation used extensively in combinatories, analysis of algorithms and various other subjects.

(Refer Slide Time: 00:50)



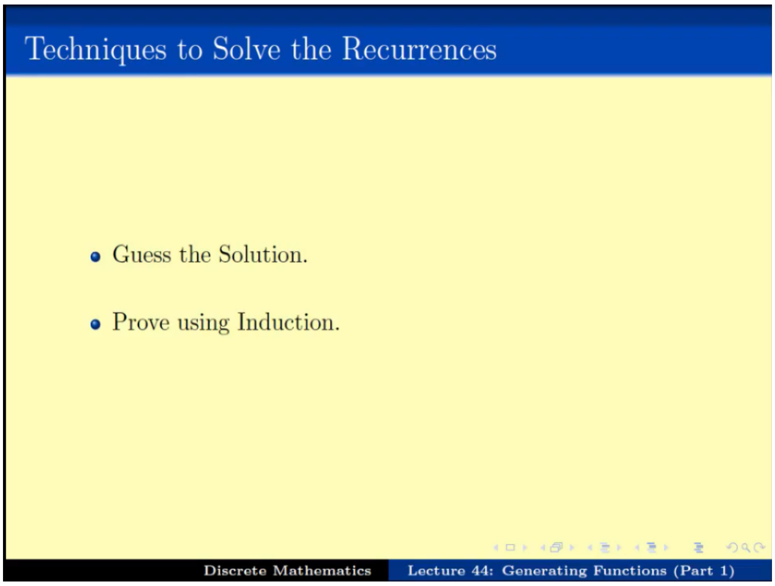
We have seen that recurrence relations can be used for modeling problems particularly counting problems and the question now is that how do we solve recurrence relations.

(Refer Slide Time: 01:04)



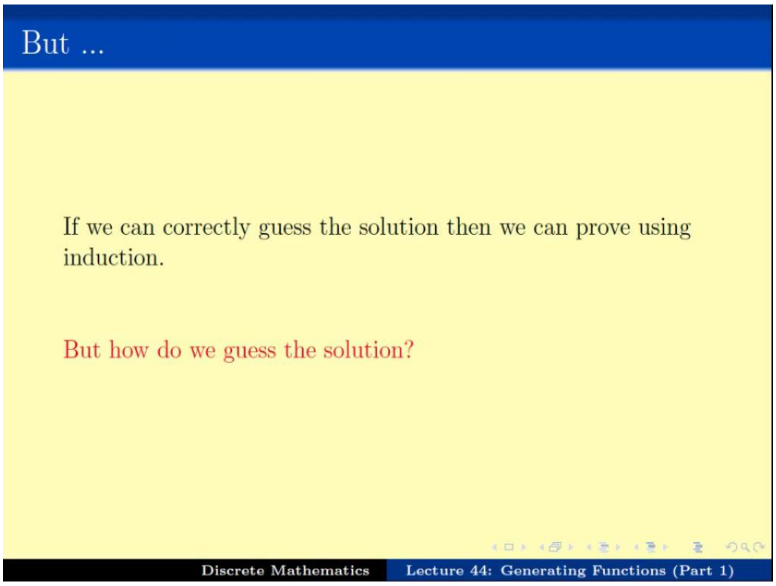
So we have seen a few of the examples of recurrence relations and for some of them we have seen techniques of solving it. Now will quickly recap the technique that we have.

(Refer Slide Time: 01:23)



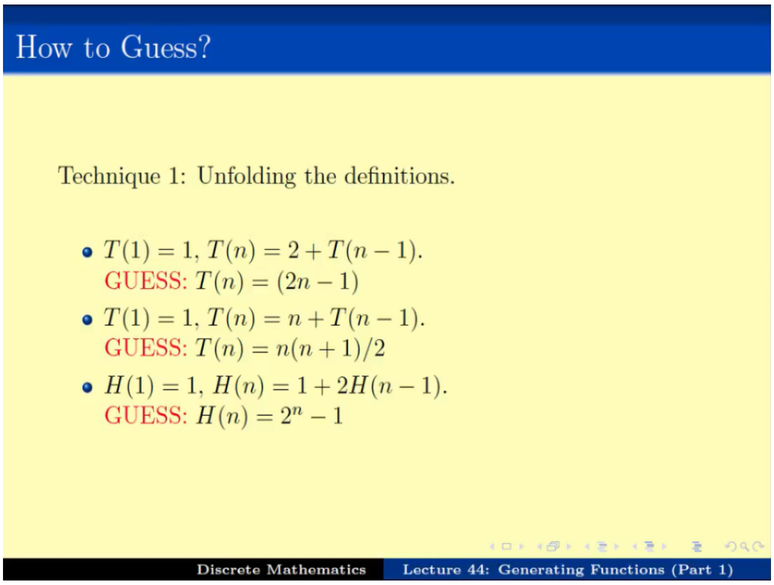
The first one was you first guess the solution and then prove using induction.

(Refer Slide Time: 01:31)



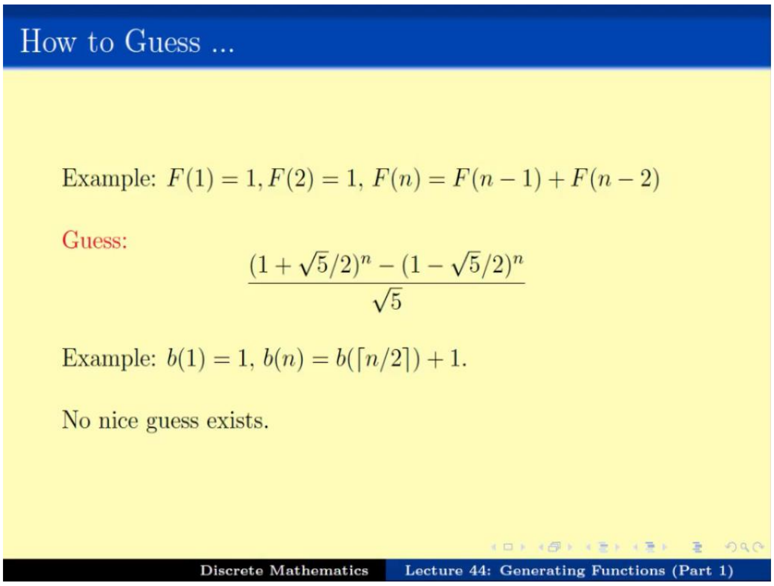
Now this technique works perfectly if you can guess the solution correctly. Once you guess the solution correctly proving it by induction is quite a simple step. But sometimes the question is how do you guess the solution?

(Refer Slide Time: 01:50)



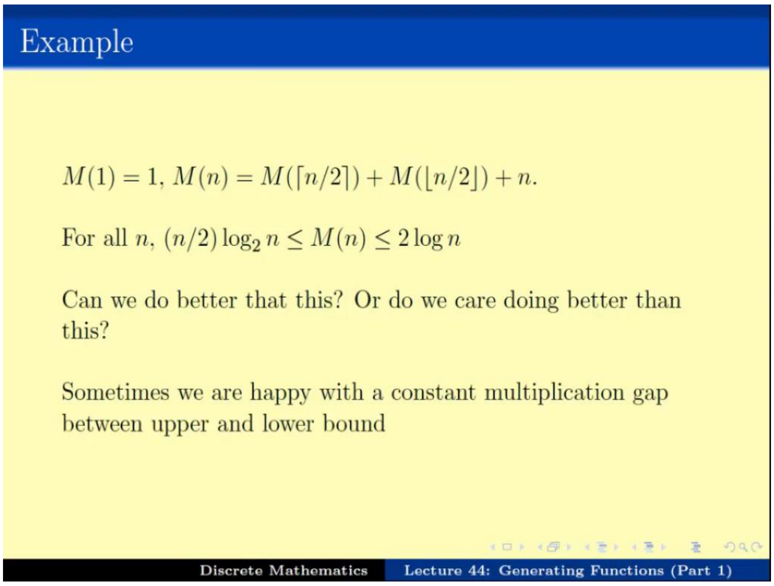
The technique one that we did for guessing the solution was by unfolding the definition and we saw how by unfolding the definition one can then try to guess the solution correctly.

(Refer Slide Time: 02:08)



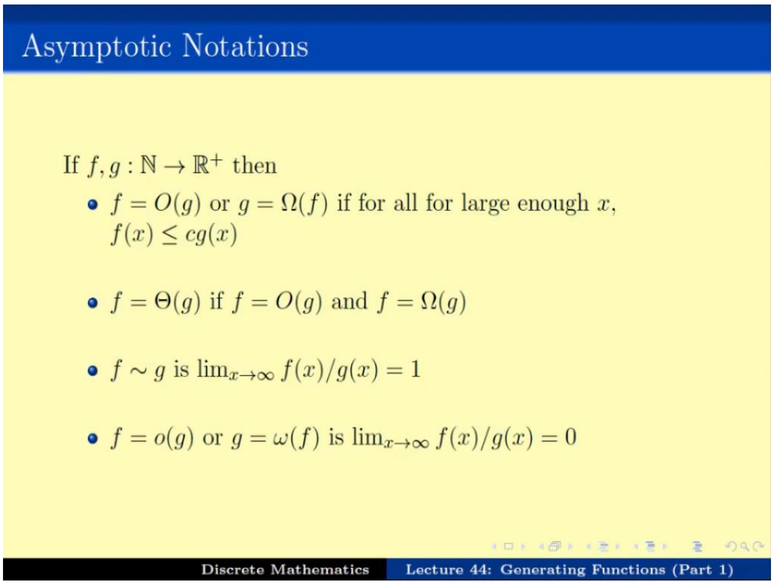
But then there are some particular class of functions or recurrences where guessing the solution is not easy. For example, this particular expression for the Fibonacci Sequence, well you cannot guess it mainly because I am telling you the final form of it which is this. And there is no way we can guess it or understand it by unfolding the definition. Similarly, we have the other techniques where we have some more complicated expressions which does not yield a very nice formula that can be guessed.

(Refer Slide Time: 03:08)



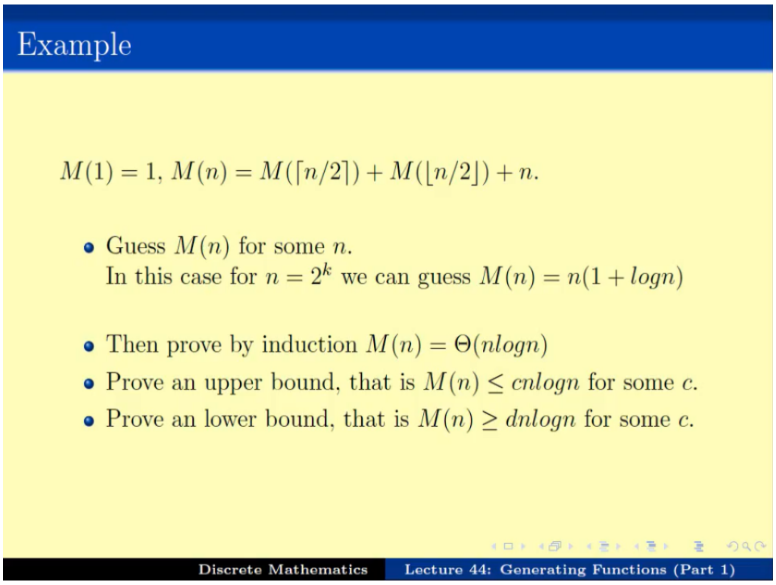
In the second case when there does not exist a very nice formula what we told is that many times we can possible come up with some upper bound environment. And once we can come up with the upper bound and lower bound we can use various notation as in for the computations to basically solve the compact form for it which in many times is good enough for us.

(Refer Slide Time: 03:30)



So we have seen the rotations of big O, big omega, theta seem small o and small omega and this is not only, this particular way of comparing functions is not only useful for solving recurrence relations but is also used for other things also.

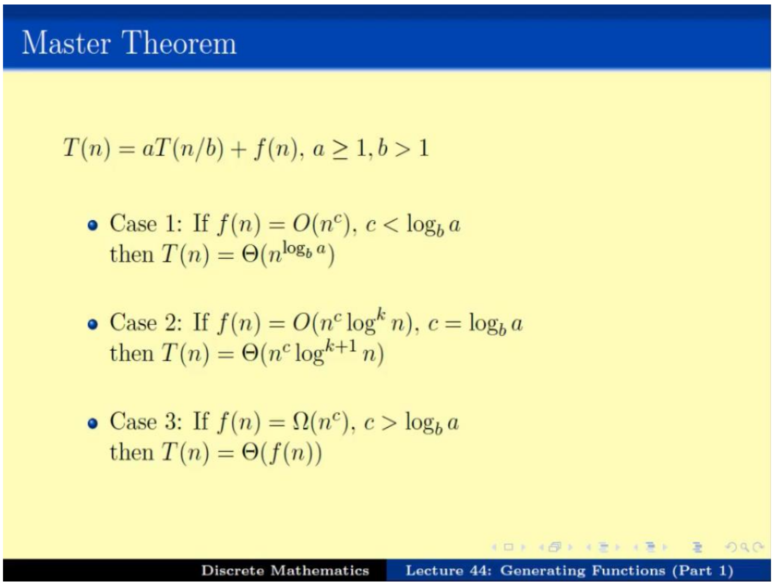
(Refer Slide Time: 03:54)



And using this, we could solve examples like this where we do not have any nice solution but we can come up with some nice upper bound and lower bound. The technique there was you first guess the Mn for some nice enough integers m, in this case for power of k and then the induction proves that this value Mn is theta of nlogn whatever you have guessed here. And it is done by first proving an upper bound and then proving a lower bound.

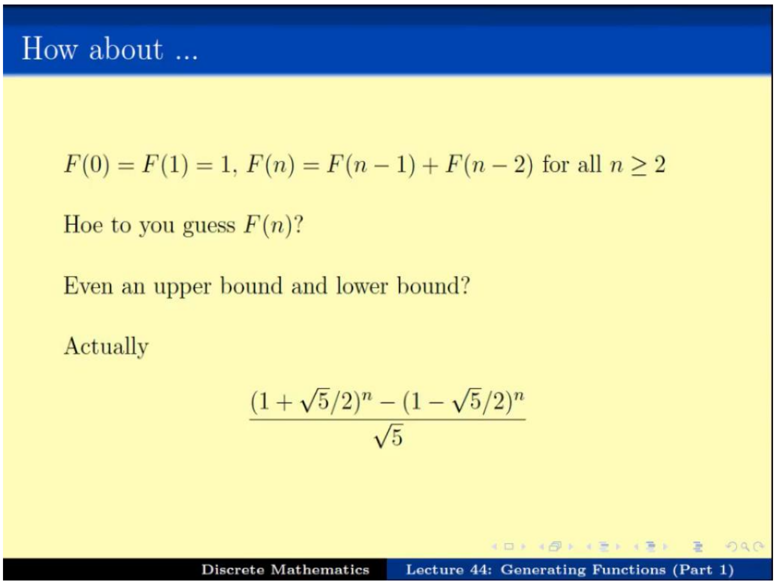
So this is the second technique that we have. First technique when you can guess formula for the recurrence exactly that we can do possibly by unfolding the definition. Second technique was when we cannot guess it exactly but we can prove a theta notation or some big O notation and so on which indirectly means that we can prove a upper bound and a lower bound.

(Refer Slide Time: 05:07)



And we saw that there is some master theorem which can help us to guess the solution easily for us depending on some of the things.

(Refer Slide Time: 05:19)

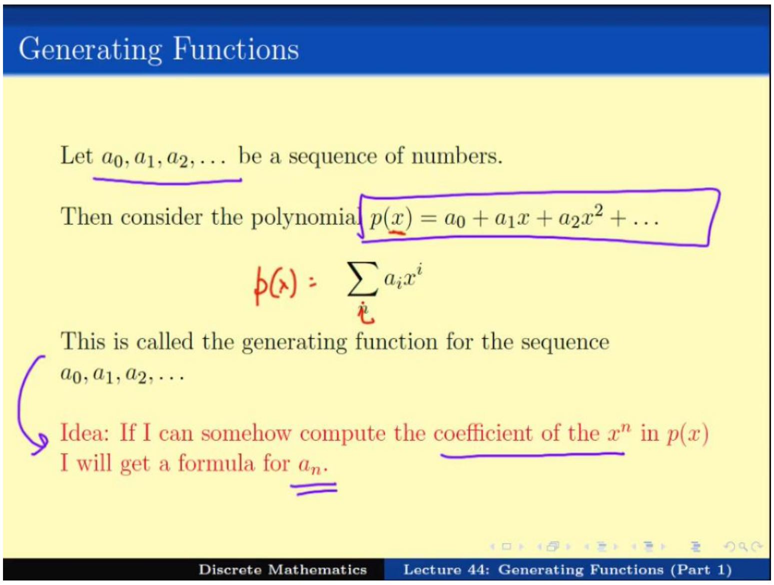


But still we have certain recurrences for which we still do not know how to false them. For example, this one, the Fibonacci number, F0 equals to F1 equals to one and Fn equals to Fn-1 plus fn-2. Now how do you guess the Fn. How do you even come up with an upper bound or lower bound and in fact I have told you, there is something I am telling you beforehand that this function finally does come down to this expression.

Fn equals to one plus square root five by two power n minus one minus square root five by two power n by square root five. Just by looking at the expression, you can imagine that this is not a trivial or not an easy recurrence solution to solve. In the next set of video lectures we will show you how to attack this problem and by doing so we will come up with a very generic technique of solving this particular recurrence.

As I told you in the beginning of this video, it is a very powerful technique for proving it and possibly a bit complicated technique also. So we will go a bit slow here.

(Refer Slide Time: 06:48)



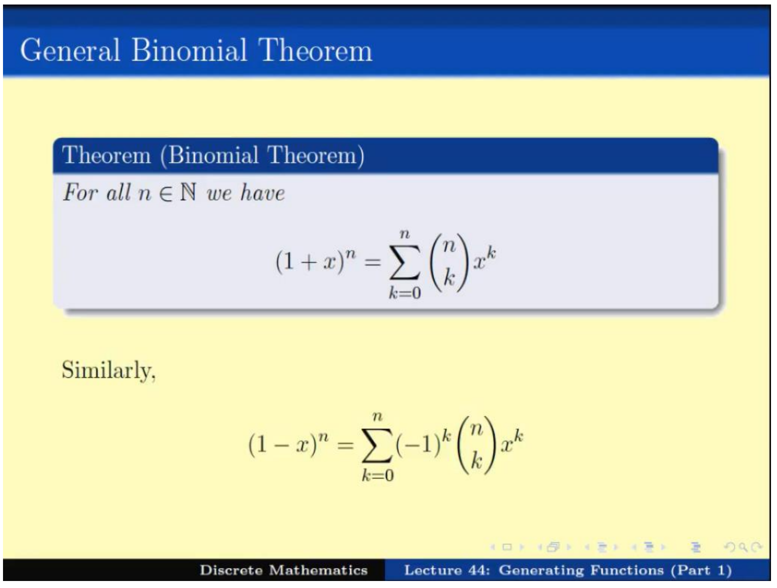
Now for this thing to work we will define what we call as generating functions. So let us start with a sequence of numbers, a0, a1, a2, till whatever a infinity or some value. Now the generating function is basically this polynomial that we define where x is a variable and p of x is a polynomial defined as a0 plus a1x plus a2x square plus a3x cube and so on. So in other words, p of x equals to summation of a1 x power I sum for all i.

Now this is the what we call as the generating function for the sequence. Now this does not solve anything. I am just representing this generating function as a polynomial. But maybe using some nice tricks that we will see in the next video, we might be able to somehow compute the nth coefficient meaning the coefficient of x power n in this polynomial px. And if I can do that then I will understand that the coefficient of x power n is nothing but an.

So I will get a formula for an which in fact would be the solution for the recurrence relation. Now this state is clearly not very obvious, how do you get it? I have defined a polynomial which at this point is nothing but an abstract polynomial because I do not know this a0 to a infinity all of them, right? If it is given in the recurrence solution all I know is the initial set and the final in some and the a in terms of the earlier ones.

But abstractly I can think of this polynomial and the goal will be to somehow get this polynomial and understand the coefficient, the nth coefficient of x power n in this polynomial. Now this is the overall idea. We will see the application of this one in the next video. In this video, we will take a small (()) (09:41) to see something what we call as a generalized binomial theorem. We will need it for what we will do next class.

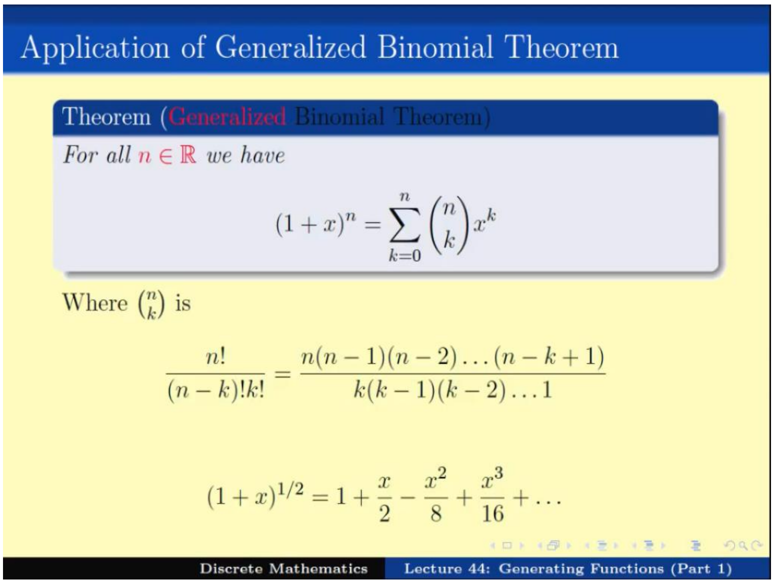
(Refer Slide Time: 10:03)



So let us start with the binomial theorem. We have done it during the time of counting and we have basically this statement that for all n, we have one plus x power n equals to sum over k equals to zero to n, n choose k x power k. Now the important thing is that what is n choose k, okay? I will come back to sum of that. So the important thing is that n is a natural number. So like zero, one, two, three, four, five and so on.

In the binomial theorem I do not know what it means when n is equals to minus one because this number n choose k is not defined for n negative or if n is not infinity here. Of course from this binomial theorem I can put other values in x and I get something like this, one minus x equals to sum over k equals zero to one, one minus x power n equals to summation k equals to zero to n minus one power k, n choose k x power k. But again I need the fact the n is a natural number, right?

(Refer Slide Time: 11:48)



Now, let us try to understand what does it mean by the n choose k. So the n choose k is nothing but n factorial by n minus k factorial times k factorial. Which means that, so n factorial is of course n multiplied by n minus one multiplied by n minus two till one. n minus k factorial is n minus k multiplied by n minus k minus one and so on till one.

So if I divide this n factorial by n minus k factorial I get something like this, n multiplied by n minus one multiplied by n minus 2 till n minus k plus one divided by k factorial which is k multiplied by k minus one multiplied by k minus 2two till one. Now if this is the definition of n choose k, let me tell you a pretty ridiculous looking theorem which is known as the generalized binomial theorem and this is that.

Even for n equals to n in real number I have the same expression, one plus x power n equals to k equals to 0 to n n choose k x power k where now this n is not necessarily natural numbers as you define what is choose k is? Well, n choose k is not this, but this. Why it is not this because is n is not natural number, where n equals minus n, the n factorial does not make any sense. But this makes sense.

I can define any first natural number I know what n is, what n minus one is, what n minus two is, I mean it go on till n minus k plus one. So this is in fact a proper definition of n choose k. So in other words I claim that there is this theorem called generalized binomial theorem where for all n which is any natural number, positive or negative I can prove one plus x power n equals to summation of k equals to zero to n, n choose k x power k where n choose k is defined as n multiplied by n minus one multiplied by n minus two till n minus k plus one divided by k multiplied by k minus one multiplied by k minus two till one.

Now let me leave it to you guys to convince yourself that this statement is true. For people who are interested in getting the proof of this, I encourage you to take a look at internet or solve it yourself. The proof is not hard; it can be done using induction. So in fact I will say that prove this theorem when n is not the integers, meaning when n equals to minus one, n equals to minus two and so on and you prove this statement.

And you will see that you will be able to prove this statement. For n equals to integers, then you have to prove for n equals rational and then finally you have to prove n equals to n is (())(15:38). So I will not give you the proof of this statement, but I will show you how this statement can be applied to get some outstanding nice things. So this is the statement that we had, you can of course strike out this thing, so that is not what the n choose k is this.

Now let us try to see what happens to one plus x power minus one. Now to understand it, we have to first understand what happens to the first coefficient, right? So this is the k equals to zero and the k equals to zero, then what is n choose k. So the n choose k is how many terms are there, there will be k of them, if k is zero the bottom one is zero factorial which is one and the top one will be nothing that is one, so I get one here.

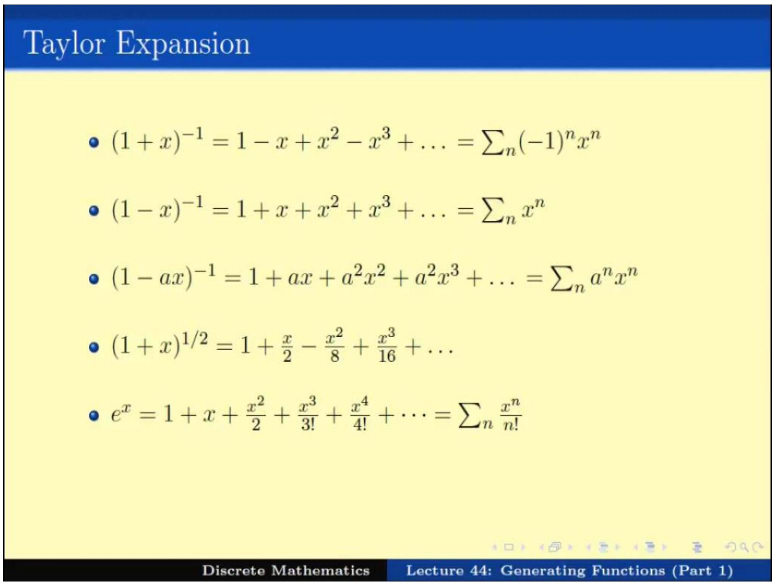
What is the coefficient of x? Coefficient of x, as k equals to one, so the bottom one will be one factorial and top one will be minus one multiplied by minus one minus, can I have anything else? No I just can have only term, so I will just have this minus one. So the next one will give the nothing but minus x or in the next one, k equals to two, then the bottom one is two factorial which is two and the top one is minus one multiplied by minus one minus one which is minus two which is of course say sum them up it is one.

So this will be plus x square and so on and you can see that what you will get finally is that, one plus x power minus one is one minus x plus x square minus x cube and so on which is of course summation of what n minus one power n x power n. Very similarly if I can, here only

I can replace x with minus x I can get one minus x minus one as one plus x plus x square plus x cube and so on which is this.

So in other words this generalized binomial theorem is helping me to write down something like a polynomial like one plus x power minus one or one plus x equals to minus one as a polynomial over x without any reciprocal. This is what is known as the Taylor series.

(Refer Slide Time: 19:17)



Similarly, for one plus x power minus two we can write down an expression like this. This is what is known as Taylor extension or Taylor series and we have the Taylor series for quite a number of them as we just now saw one plus x power minus one is one minus x plus x square minus x cube and so on.

Similarly, one minus x is one plus x plus x square and so on. From this we can also get something like this form, one minus ax as one plus ax plus a square x square plus a cube x cube plus so on. We can also have something for like one plus x power half and they have Taylor series for things which are not polynomial like e power x which is one plus x plus x square by two which is basically some of our x power n by n factorial.

So these are called the Taylor series expansions for functions which are not necessarily polynomials. So most functions that you have can be written as a polynomial or in the Taylor series expansion. We have used this particular idea to see how we can use the generating function for solving things like the Fibonacci number recurrent and so on. We will do it in the next class. Thank you.