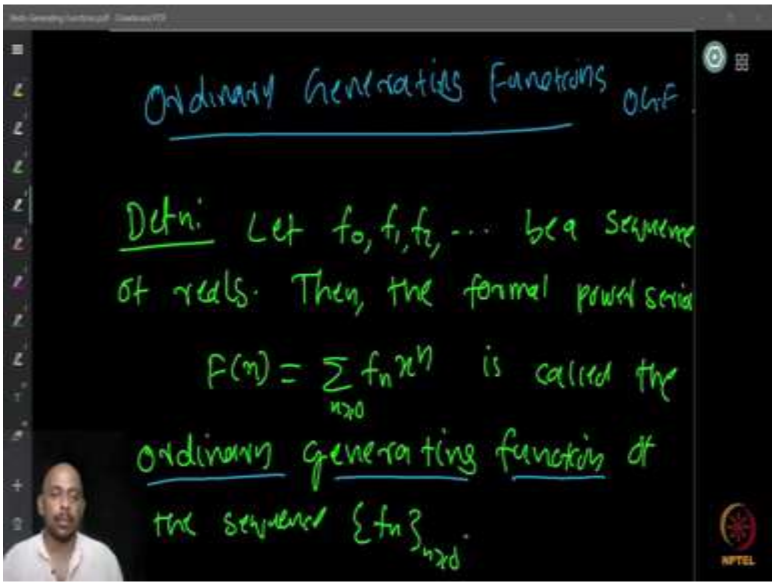
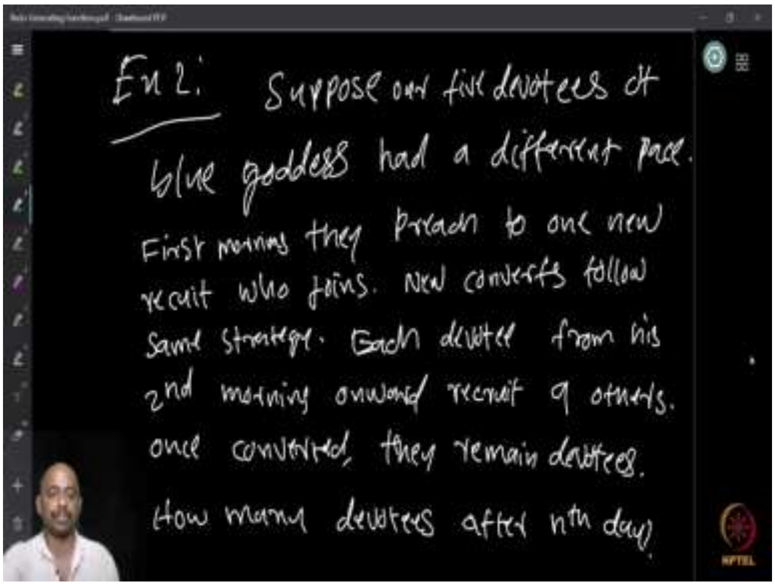
Combinatorics
Professor Doctor Narayanan N
Department of Mathematics
Indian Institute of Technology, Madras
Product of Generating functions

(Refer Slide Time: 0:20)



In the previous lecture, we looked at ordinary generating functions and then one example. Now, we will look at a couple of more examples before going into further topics. So, let us start with another example.

(Refer Slide Time: 00:57)



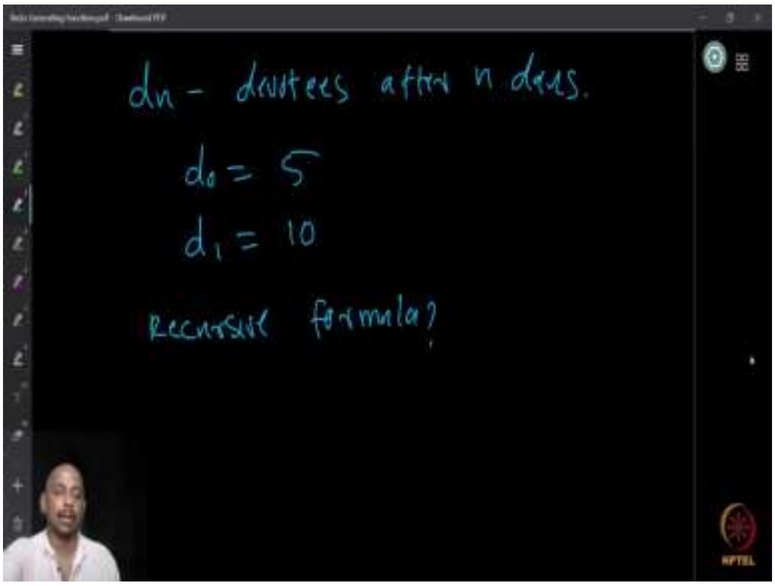
Here is our example 2. Again we have our five devotees of the blue goddess. What they do is that, they change their strategy of conversation. They started converting in a different way. What they do is that, on the first day they will preach to one recruit, who joins. So, the first morning right after the religion was formed, the first morning, they will convert only one person.

Now, the, new converts follow the same strategy, once they join the religion, the next morning, they will preach to one person, then recruit. Then every devotee from his second morning onwards recruits nine others, so they have increased the base. In the first day, it will be relaxed, but then, next day onwards, they are more energetic and they are going to convert 9 people every day morning.

Once converted all the converted people remain the devotees, that nobody changes their opinion. Now, in this case, how many devotees will be there after the n'th day. So, a slight modification from the previous question. We started with our 5 devotees of blue goddess who wears blue t shirts, and then they convert people in the following way.

First day, they will preach to only one person, second day onwards they will introduce 9 more persons and who ever was converted their next morning, the first day they will convert one person and then the next morning onwards they will convert 9 others. Well, similar to the previous question, we can form a recurrence relation and then try to solve it. Using the generality functions, let us see how we can form the recurrence relation. I recommend that you really try to do it on your own, then once you give it a try, come back here and continue.

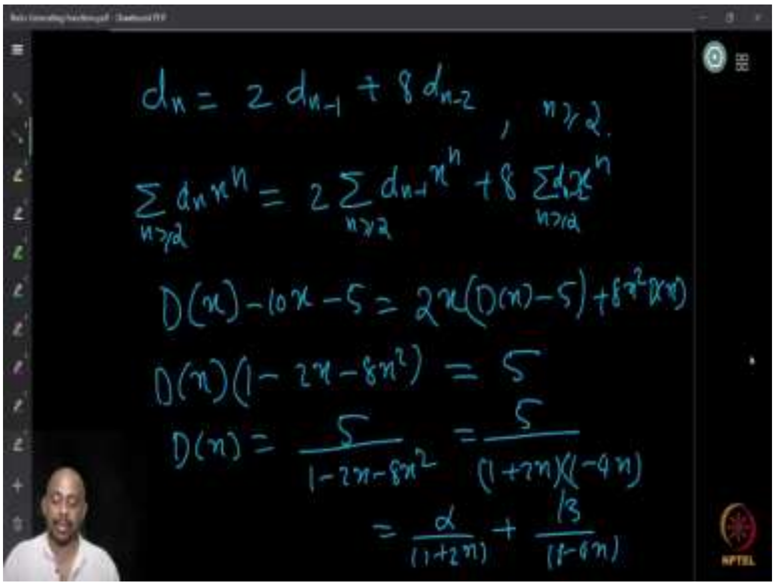
(Refer Slide Time: 03:18)



So, we have the following observation that if d_n we denote the devotees after n days, then $d_0 = 5$, at the 0'th day, when the religion is formed, there are 5 devotees. And at the end of the first day you have 10 devotees because they converted, each of them converted 1 more person, so I have 10 person.

Now, nobody is going to give up the religion after they join it. Therefore, at the end of day 1 you have 10. Then what is the recursive formula. So, we have $d_0 = 5 \; and \; d_1 = 10$. Now, can you find out a formula for d_n, where n is greater than equal to 2. So, if you think about it, you will see that. So, the first day they converts 5, but second day onwards each of them converts 9 others.

(Refer Slide Time: 04:22)



So whoever is present there they will all convert 1 person at least. So everybody converts 1. So, all the d_{n-1} people will convert at least one person. So, at the end of the day you have $2d_{n-1}$ members.

Now, if each person who was there in the previous, so this is not the first day for people, they will all converts 8 more people, because in total they convert 9, so 1 is already counted. So, there they convert eight more people. So, people who are the second day which is d_{n-2}, each of them, where d_{n-2} of them are having the second day now, so they will all convert 9 people.

So, we have $d_n = 2d_{n-1} + 8d_{n-2}$, because whoever is there then everybody converts one persons and then each of the persons who is having the second day or more, all those people will convert 8 more others.

Thus, we have the recurrence relation. Now, we can use the method of generating functions. So, how do you do that, for that we multiply by x^n on both sides of the recursion relation and then some overall n greater than or equal to 2. So,

$$\sum_{n\geq 2} d_n\, x^n \;=\; 2\sum_{n\geq 2} d_{n-1}\, x^n \;+\; 8\sum_{n\geq 2} d_{n-2}\, x^n$$

Why n greater than or equal to 2 because the recursion formula holds for n greater than or equal to 2 and we have the term d_{n-2}. So, the index only starts from zero. So, I have n is greater than or equal to 2. Now, if I denote by $D(x) = \sum d_n x^n$.Then the first two terms is mean $d_1 x$ and d_0.

So, we subtract that to get the LHS, $D(x) - 10x - 5$. On the other hand, on the right side, you have $2\sum_{n\geq 2} d_{n-1}\, x^n$. I take x outside I get $2x(D(x) - 5)$ because the first term is missing from the summation. And then you have, from the last one you have you can take x^2 outside you will get $8x^2 D(x)$.

Therefore I get the formula now,

$$D(x) - 10x - 5 = \; 2x(D(x) - 5) + \; 8x^2 D(x)$$

simplify this for D(x) I get,

$$D(x)(1 - 2x - 8x^2) = 5$$

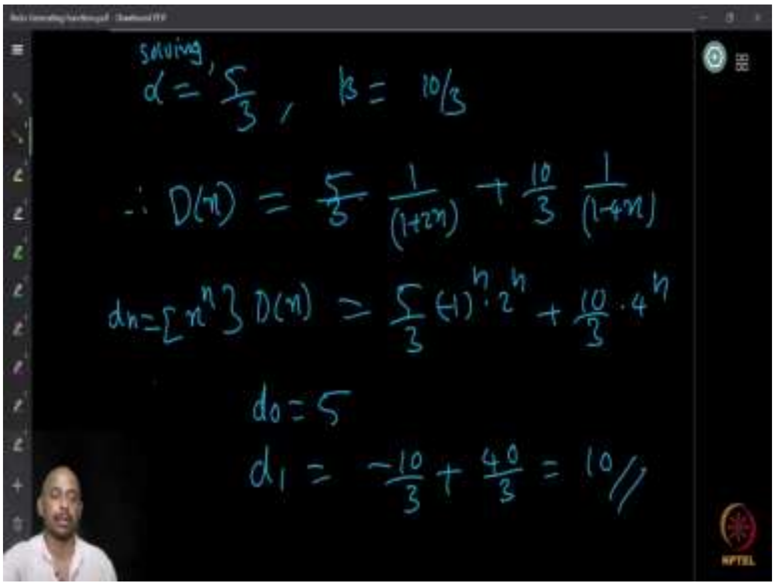
$$D(x) = \frac{5}{1 - 2x - 8x^2} = \frac{5}{(1 - 2x)(1 - 4x)}$$

So, this will be written as a method of partial fractions, we can write it as

$$D(x) = \frac{\alpha}{1 - 2x} + \frac{\beta}{1 - 4x}$$

So, we solve for alpha and beta we get

(Refer Slide Time: 08:55)



$$\alpha = \frac{5}{3}\,, \beta = \frac{10}{3}$$

Now,

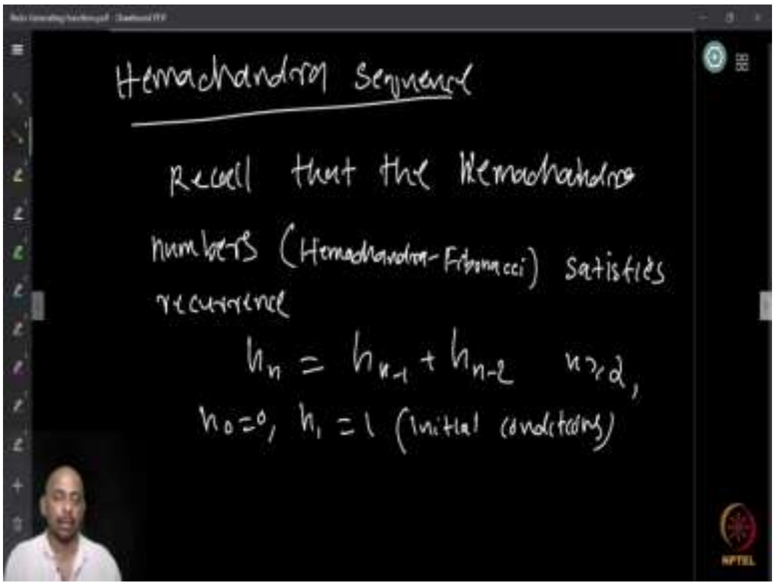
$$D(x) = \frac{5}{3}\frac{1}{1-2x} + \frac{10}{3}\frac{1}{1-4x}$$

Now, this is now easy because coefficient of x^n in D(x), can be find out by expanding these two functions to series. So, I will get this as

$$d_n = [x^n]D(x) = \frac{5}{3}(-1)^n 2^n + \frac{10}{3}4^n$$

So, I get a close formula for dn. So now, we see how powerful this method is.

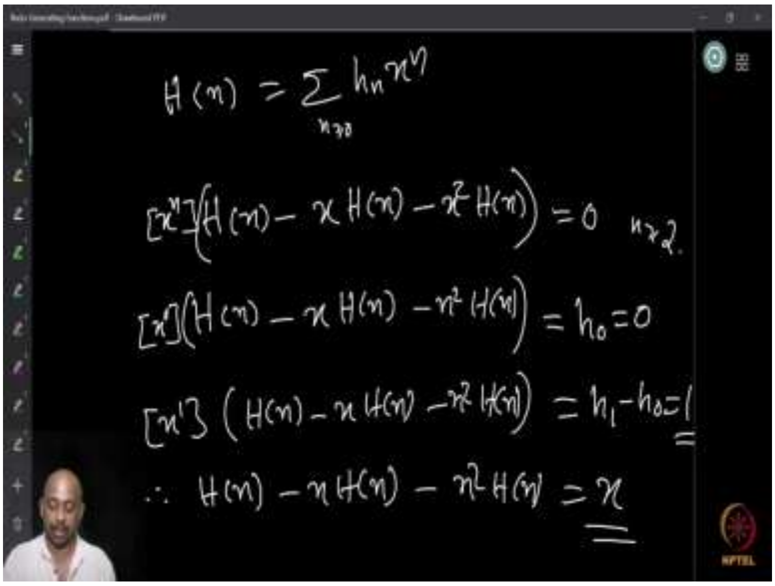
(Refer Slide Time: 10:53)



We look at another famous sequence, which we call the Hemachandra sequence. So, Hemachandra sequence if you remember, is also called the Fibonacci Hemachandra or Hemachandra Fibonacci sequence and this sequence satisfies the recurrence relation as we already know is $h_n = h_{n-1} + h_{n-2}, n \geq 2$. So, starting conditions, initial condition can be $h_0 = 1, h_1 = 1$, and then you can continue.

I mean you have a slightly different way of defining the initial conditions because sometimes people start by 1, then 2, and then etcetera, 1,1, 2 etcetera, sometimes it will start with 0, 1 etcetera. So, this will make only a very minor difference, because we know just shifting in the index. So, we do not have to worry about that, we will assume that this is our current initial condition. So $h_n = h_{n-1} + h_{n-2}, n \geq 2$, $h_0 = 1, h_1 = 1$.

(Refer Slide Time: 12:12)



Now, let us compute this H(x) in a slightly different manner. We know that $H(x) = \sum_{n\geq 0} h_n x^n$. Now, let us look at the coefficient of x^n in $H(x) - xH(x) - x^2H(x)$. Now, why do I look at this because $h_n = h_{n-1} + h_{n-2}$, so, that identity is there.

Now, if I look at coefficient of x^n in H(x), that is basically h_n then coefficient of x^n in xH(x) is basically h_{n-1} and coefficient of x^n in $x^2H(x)$ is h_{n-2} . So, this is clear because x^2 shift the terms by 2.

Now, $h_n - h_{n-1} - h_{n-2} = 0$, because for n greater than or equal to 2, we have this identity satisfied by the Hemachandra sequence. So

$$[x^n]\big(H(x) - xH(x) - x^2H(x)\big) = 0\ , n \geq 2$$

Now, for n is equal to zero, what is the coefficient of x raised to 0? Coefficient of x raised to 0 in this is precisely h_0.

$$[x^0]\big(H(x) - xH(x) - x^2H(x)\big) = h_0 = 0$$

Now, coefficient of x raised to 1 in the same is that,

$$[x^1]\big(H(x) - xH(x) - x^2H(x)\big) = h_1 - h_0 = 1$$

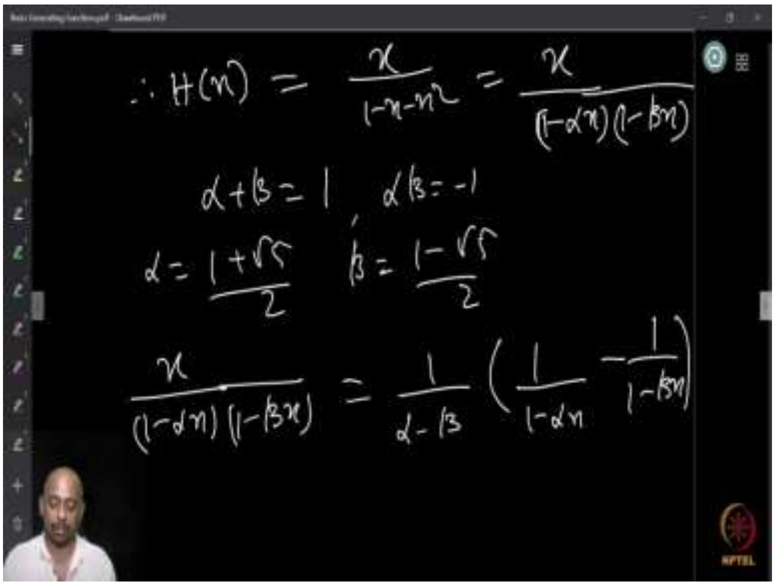
So therefore, we have the complete information about the difference of this power series. So,

$$H(x) - xH(x) - x^2H(x) = x$$

So the equation written in x which is with coefficient 0 for whenever the power of x is at least 2, coefficient 0 whenever the power is 0, and coefficient 1 when the power is 1, so that is the polynomial x. So, therefore, we get this difference of the formal power series, these three is precisely x. So, now I get a formula

$$H(x) = \frac{x}{1 - x - x^2} = \frac{x}{(1 - \alpha x)(1 - \beta x)}$$

(Refer Slide Time: 16:21)



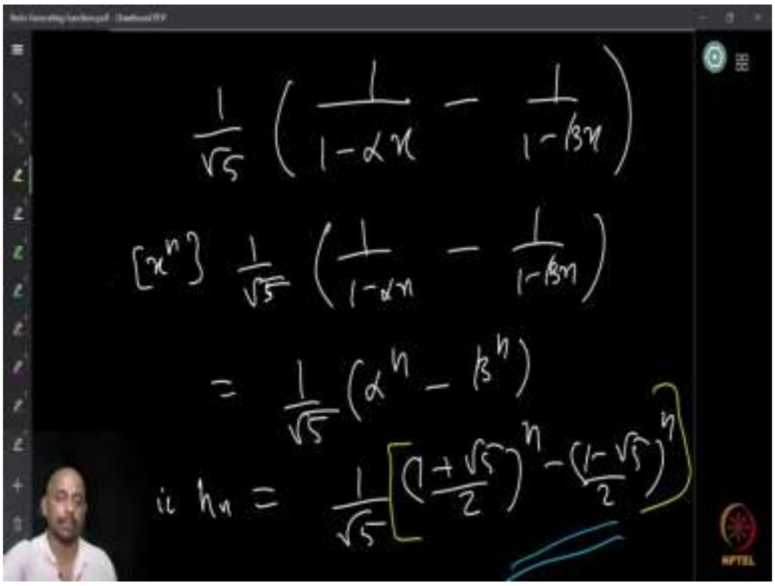
We get $\alpha = \frac{1+\sqrt{5}}{2}$, $\beta = \frac{1-\sqrt{5}}{2}$

So, we can immediately get it. And the roots alpha and beta, and once you have this, we use the method of partial fractions. So, once you write down, you can show that this reduces to the following, I can write it as. So, this computation now is routine. So, therefore, I leave it to you, I do not want to spend time on working out this minor details.

So, you work with this and if you have any questions, get back to me. So

$$\frac{x}{(1 - \alpha x)(1 - \beta x)} = \frac{1}{\alpha - \beta} \left(\frac{1}{1 - \alpha x} - \frac{1}{1 - \beta x} \right)$$

(Refer Slide Time: 17:45)



So, if you solve it like this, then

$$\frac{1}{\alpha-\beta}\left(\frac{1}{1-\alpha x}-\frac{1}{1-\beta x}\right)=\frac{1}{\sqrt{5}}\,(\frac{1}{1-\alpha x}-\frac{1}{1-\beta x})$$

so now we need to just find out the coefficient of x^n, in this

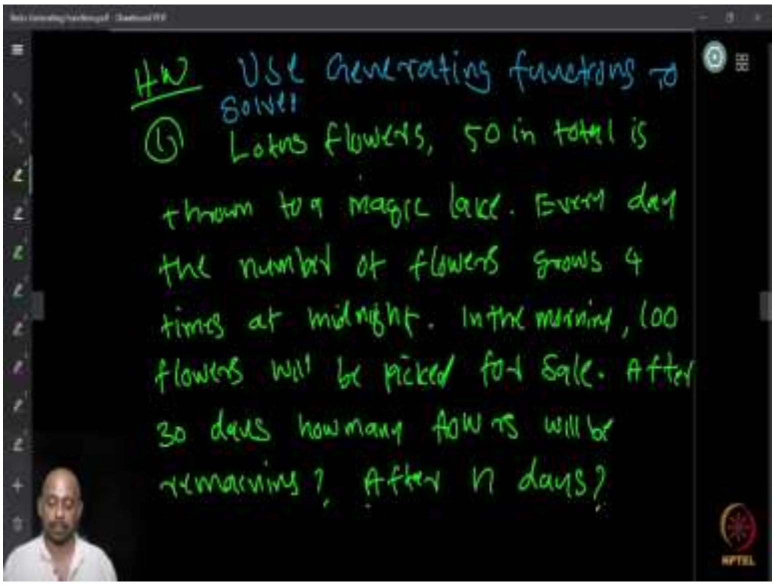
$$[x^n]\frac{1}{\sqrt{5}}\left(\frac{1}{1-\alpha x}-\frac{1}{1-\beta x}\right)=\frac{1}{\sqrt{5}}(\alpha^n-\beta^n)$$

So,

$$h_n=\frac{1}{\sqrt{5}}[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n]$$

Now, it may be interesting to see that this is always an integer, because h_n is our Fibonacci or Hemachandra sequence.

(Refer Slide Time: 19:33)

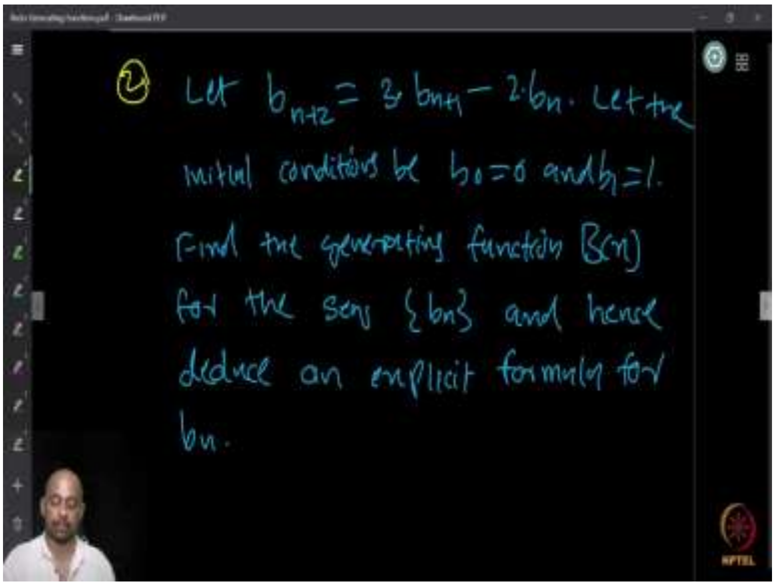


I will give you a couple of homework questions. Use generating functions to solve the following. We have the following situation. So, there is a nice pond or a lake, let us say it is a magic lake. So, this magic lake has this property if you throw let us say 50 lotus flowers to the lake, then every day, the number of flowers grows by 4 times at the midnight.

So, every night the number of flowers multiplies by 4. So, if there are 50, then next morning, you will see 200 of them. So, in the morning 100 flowers will be picked for sale. So, people use this that you put 50, or whatever, let us say some flowers and next day you have 4 times, so, you take some of them and then sell it.

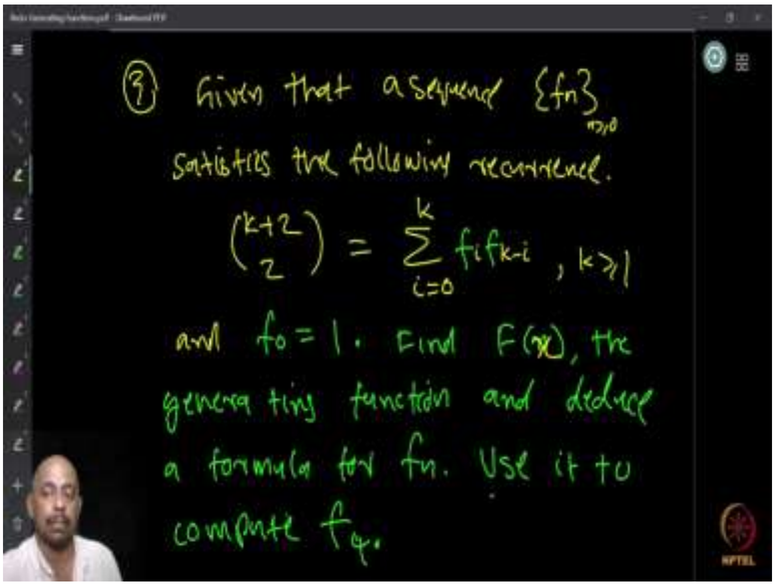
So, if I start with 50, then the first day I get, first night I will get 200, then I pick up 100 and sell it, but then the remaining 100 will be there that will multiply by 4 times, so 4 into 400 then I again take 100, you have remaining 300 and then that is this way I keep on doing. Now, after 30 days, how many flowers will be there in the lake and how many will be there after n days in general. So, find the recursion formula and solve it, using the method of generating functions.

(Refer Slide Time: 21:17)



Second question ask you to solve the recursion relation $b_{n+2} = 3.b_{n+1} - 2.b_n$. Now, the initial conditions are assumed to be $b_0 = 1, b_1 = 1$. Now, again use this information find the generating function B(x) which is the generating function for the sequence $\{b_n\}$. And using this deduce an explicit formula for b_n. From the generating function, we want to find a close formula for b_n.

(Refer Slide Time: 21:53)

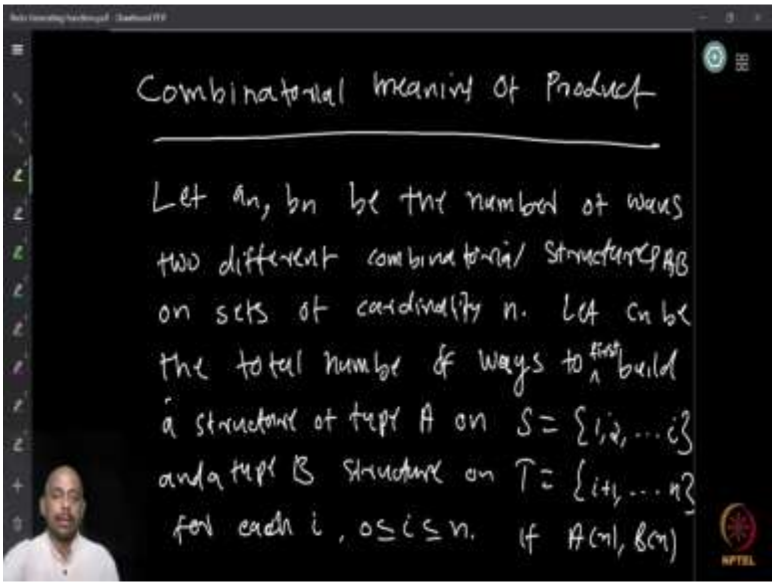


That question ask you to do the following, given a sequence let us say $\{f_n\}_{n \geq 0}$ and this f_n satisfy the following recurrence relation

$$\binom{k+2}{2} = \sum_{i=0}^{k} f_i f_{k-i}, k \geq 1$$

Now, find F(x) the generating function for the sequence f_n and then use it to deduce a formula for f_n and then compute f_4 using this identity, so this is what we want.

(Refer Slide Time: 22:53)



Now, we look at the combinatorial meaning of the product of generating functions. So, suppose you are given two sequences, let us say a_n and b_n. Which represent the number of ways, let us say two different combinatorial structures on sets can be built on an n-element set. So, given an n-element set I want to make let us say permutation.

Similarly, given n as a parameter, I want to say what is the number of triangulations an n-sided polygon can have. So, these kinds of things can be represented by sequences a_n and b_n.. Now, we define c_n to be the total number of ways. So, given an n-element set, let me take, so let us say that you have this set 1 to n, there is an ordering we fix. And then on this set let us say 1 to n, what I am going to do is that I want to first build a structure of this type T. Let us say we want permutation.

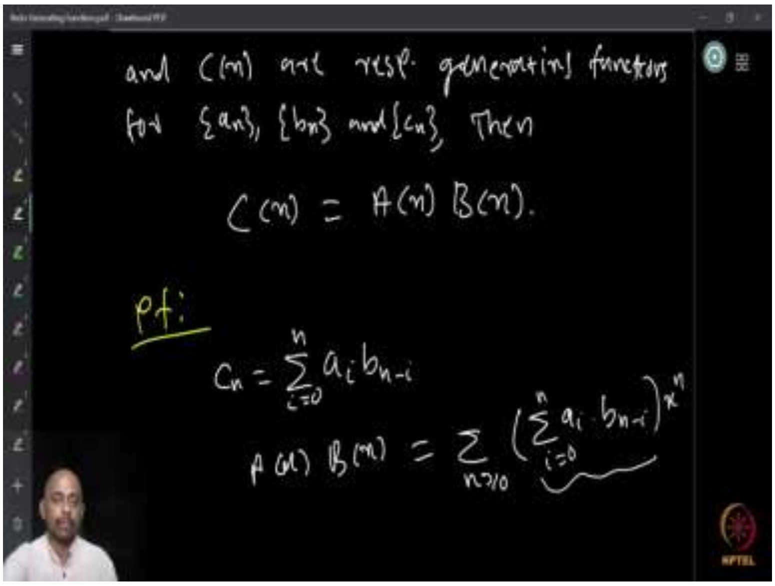
So, I want to make permutations on set, let us say 1, 2, i for some i. Then I want to make type B structure which is in our example, this was we were looking at what the triangulations of the polygon. I want to count the polygons and the number of such triangulation. I want to make triangulation of a polygon which can be made out of i +1 to n or like I want to put a structure on the set T is equal to {i +1,…, n}.

So, we have two types of structures, let us say type A and type B, so type A such as, on an n-element that I have a_n of them, and type B such as there are b_n of them. Now, c_n is defined to be the total number of ways to first build a structure of Type A on the set 1 to i and then type

B structure on the set i + 1 to n for some i. So, for each i this can be done. So, from i ranging from 0 to n, I can do this. So, I can say that okay, I do not build anything, I put the empty set on which if I can do some structure of type A, you do that.

Then for the remaining all elements I put the structure B or I can say that for one element, 1 I will take here remaining 2 to n minus, 2 to n , I will put a structure B again. Similarly, I can take the all elements 1 to n and put structure A on that and then in the empty set I can put structure B. So, this way I do for every possible i. Now, if A(x) and B(x), represent the generating functions for an and b_n.

(Refer Slide Time: 26:57)



And C(x) represent the generating functions for c_n, then the claim is that C(x) = A(x)B(x), the product of the generating functions of a and b will give you the generating functions for c_n. And the proof is simple.

$c_n = \sum_{i=0}^{n} a_i b_{n-i}$

So, what is our definition for c_n, so we said that c_n, is the number of ways to first build a structure of type A on 1 to i. So, that is a_i. And then I can put structure of type B on i plus 1 to n which is n - i elements are there, so given any n - 1 as the parameter, I have exactly b_{n-i} structure of type B.

So, therefore, for a fixed i, I have a_i structures I can make here and b_{n-i} possibilities to do on the type B structure. So, $a_i b_{n-i}$ structures I can make. This by the product rule, so that we have learned before.

Now, i can vary, because we said that for each i, we have this option, so I can fix the i to be either 0, 1 or up to n. Therefore, i can range from 0 to n, and then each one is separate, so they are disjoint sets, so I can do the sum. So, I can $\sum_{i=0}^{n} a_i b_{n-i}$, and that is precisely the number of ways to make the C type structure. So, that is precisely c_n.

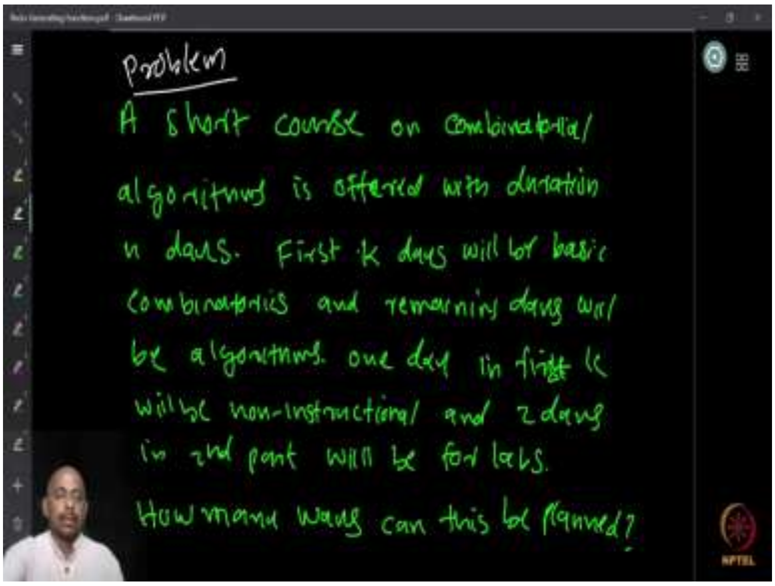
But then that is the definition of the product of the generating functions, we said that

$$A(x)B(x) = \sum_{n\geq 0} \left(\sum_{i=0}^{n} a_i b_{n-i}\right) x^n$$

This is the generative function of the product, but then what is this, this is precisely c_n.

So, therefore, I get the proof that C(x) = A(x)B(x). So, this is a way to interpret what happens when you take the product of two generating functions. And if the generating functions are representing counting sequences for structures. There could be other ways to interpret this, but this is one way that we want to do, this is a standard way to do it.

(Refer Slide Time: 31:27)

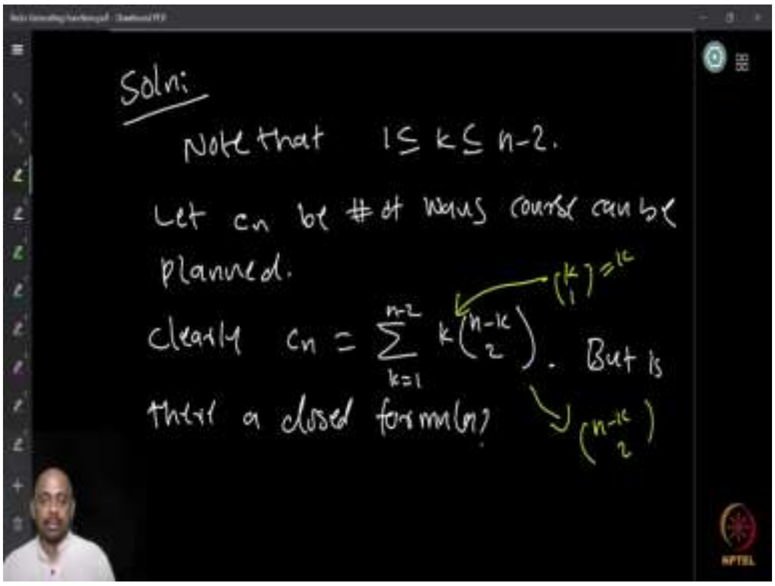


Now, we can now use this to solve problems. Here is an example. There is a short course on let us say combinatorial algorithms, which is offered with the duration of let us say n days. So, there is an n day course, like 30 day course or 20 day course or 50 day course, something like that. There is a short course on combinatorial algorithms offered and it has duration of n days.

Now, the instructor decides, okay the first k days will be basic combinatorics, the remaining days will be algorithms. Now, the n days are split into two, first k days will be basic combinatorics because we need to learn combinatorics to be able to use in the combinatorial algorithm, the first k days will be combinatorics remaining will be algorithms. Now, one day in the first k will be non-instructional.

So, out of the k days one day I will say I will not teach on one day. That is 1 day is holiday that is decided, then 2 days in the second part will be also reserved for labs, because without doing labs we do not want to finish the board. So, therefore, we will say 2 days will be labs. Now, the question is that how many ways one can plan this course. So, because k can vary, and the choice of the holiday can also vary. So, therefore, we need to figure out this.

(Refer Slide Time: 33:14)



So, first we note that, since we need to have at least 1 day as holiday, k must be at least 1 because if there is no day, I mean k cannot be 0, because if there is no day, I cannot choose one day as a holiday for the first part. So, k must be at least 1 but it cannot be more than n - 2 because if there is n days semester, 2 days must be reserved for the holidays in the second part. So, because we need at least 2 days there, it cannot be less than 2 days.

But apart from that there is no restrictions, so I can have k between 1 and n - 2. Now, let us define c_n to be the number of ways the course can be planned. So, what is c_n, c_n is precisely the number of ways to first split, the n days into two parts, like 1 to k and k + 1 to n. Then choose 1 holiday for the first part then 2 holidays for the second part. And once you do that, your course is already planned.

So for not holidays, 2 days for the labs. we have clearly c_n as splitting into k and n - k. And then from the k, I need to choose 1 day and so there is $\binom{k}{1}$ possibilities are there. So, the choice of one holiday for the first part of the course will be k choose 1 which is equal to k. And then the remaining n - k days are there for the second part and 2 days I need to do fix it for lab. So, 2 days I get $\binom{n-k}{2}$ possible ways.

So, this is a number of ways to choose 2 days out of the n - k days. So, $k\binom{n-k}{2}$, this is the number of ways to design the course. We know if you how exactly k days for the first part and n - k for the second part. Now, k can vary from 1 to n minus 2. So therefore, I can sum over all these possibilities, $c_n = \sum_{k=1n}^{n-2} k\binom{n-k}{2}$.