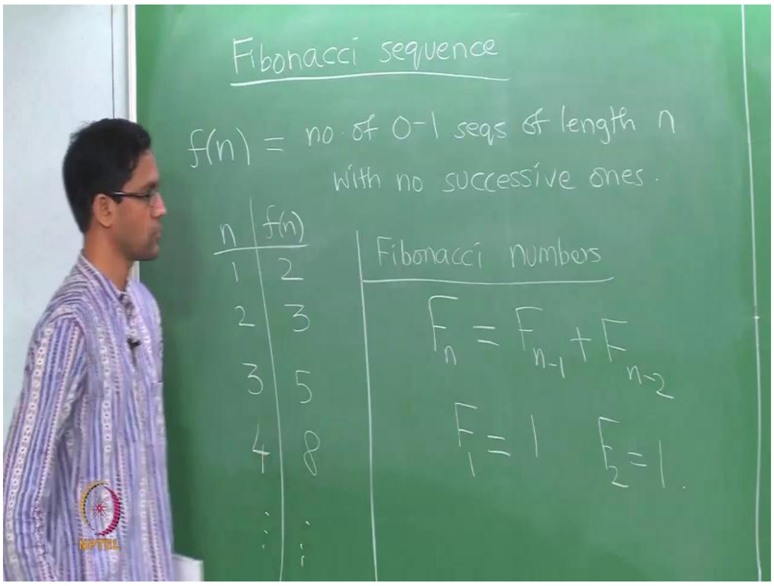
An Invitation to Mathematics
Prof. Sankaran Viswanath
Institute of Mathematical Sciences, Chennai

Unit
Combinatorics
Lecture - 12
Fibonacci Numbers; an identity and a bijective Proof

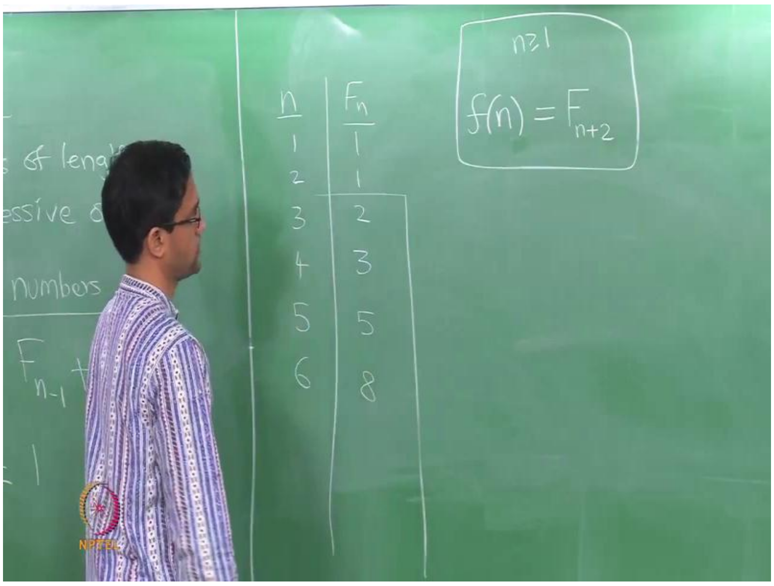
(Refer Slide Time: 00:29)



Welcome back, so last time we talked about the problem of a combinations with restrictions of counting the number of 0 1 sequences, which have no two successive ones and we found. So, we call that sequences f of n, so which is the number of 0 1 sequences of length n with no successive ones and we tabulated the first few values of f of n. So, here is what we got, so here is n and here is the various values f n. So, if n is 1 it is 2 and so on and the rule for forming the next value of f is to just take the sum of the two preceding values and the sequence here 2, 3, 5, 8 and so on is actually let us almost the famous Fibonacci sequence of numbers.

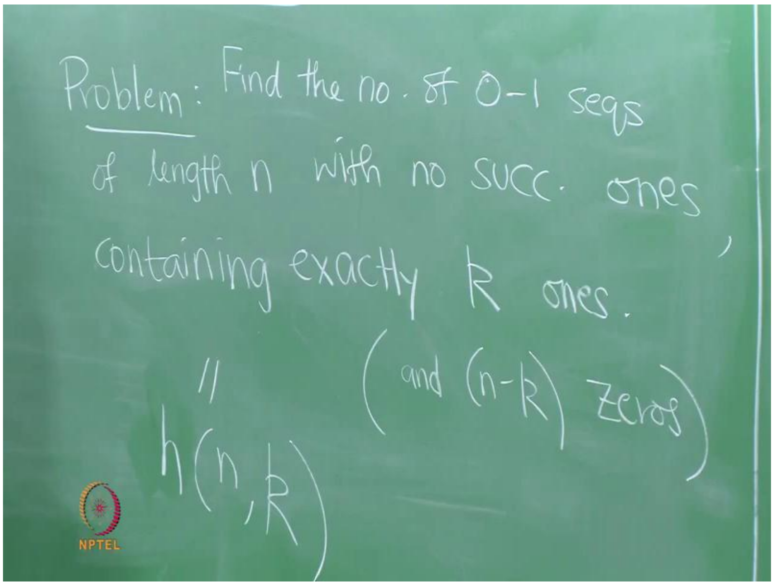
So, we will just briefly recall what the Fibonacci sequence is, so the Fibonacci numbers or Fibonacci sequences are often denoted capital F of n. So, what is capital F of n by definition, well it is given by the following two pieces of information. Firstly, it has the same recurrence that we wrote out for f of n, but the initial conditions are different. So, F of 1 terms out to be 1 and F of 2 is also 1, so it starts at 1 1 and then when you form the next term it is the sum of the two proceeding.

(Refer Slide Time: 02:08)



So, the Fibonacci sequence looks like this n verses F n, so 1, 2, 3, 4, 5, 6 and so on. So, this is 1 1 followed by 2, 3, 5, 8 etcetera. So, observe that from here on what you have are just the values of f of n. So, observe from this table it is clear that the number f of n that we are counting is just F the Fibonacci number n plus 2, so this for all n at least 1. So, the sequence that we constructed is really just the Fibonacci numbers, but starting with the third Fibonacci number which is the number 2.

(Refer Slide Time: 02:58)

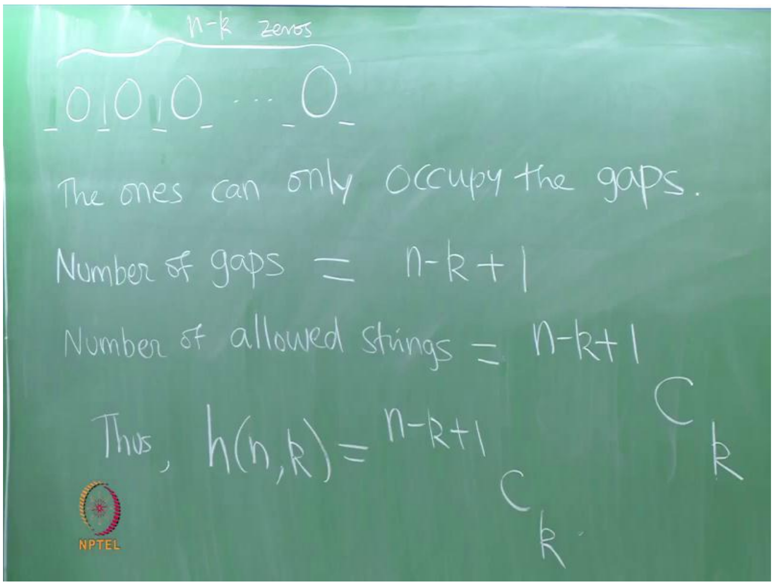


And so now, here is another interesting problem going back to our combinations with restrictions, let us count it with in a somewhat more finer way. So, let us do the following let us now just look for the number of 0 1 sequences of length n with no successive ones, let us also impose the further condition. So, find the number of 0 1 sequences of length n with no successive ones that, so far it is a same condition.

But, let us impose one further condition that it containing exactly k ones. So, we have a further parameter k and we know what our sequence to have length n and also it should have exactly k ones. So, well if it as k 1 what is it mean of course, it means that the rest of the entries are zeros and n minus k zeros, so let us give this number and name. So, this number here let us call it, may be instead of just f of n it as additional parameter k. So, there is... So, maybe should not even call it smaller.

So, let us call it the h of n comma k, so there are two parameters n and k and the value of course, will depend on both of them. So, let us try and do this computation, let us find the number of sequences, now what is that we need to do. So, this problem is actually somewhat simpler than trying to do the full problem, the original problem in the following sense. So, we know already the profile of our string we know how many zeros and how many ones it contains.

(Refer Slide Time: 05:06)



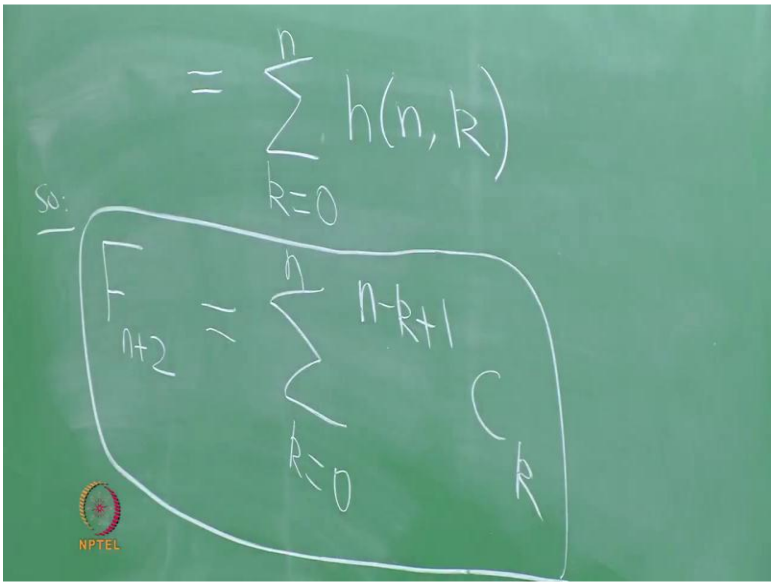
So, observe our string has n minus k zeros, so let us write down the zeros first. So, here is the string in which I am not yet written the ones. So, observe I have n minus k zeros and now let us worry about where the ones can occur. So, what we know is that the ones cannot occur in successive positions. So, which means that, any time there is a 1, then what you need to have write after it is a 0.

So, if you know ask where can be the ones be position in among the zeros. So, you just for the moment imagine that there are gaps between the zeros. So, between each successive pair of zeros imagine there is a gap. Now, the ones can occupy positions in these gaps, where can the ones leave. Well, having a 1 here is or having a 1 here is fine, here is okay, now what you cannot have is to have two ones occurring right after each other.

So, in other words we should think of there being exactly room for a single entry between two zeros, like there is just a single gap there. So, here is a way of thinking about this string, the ones can only occupy the gaps. So, in our string the ones can only occupy the gaps, now observes since there are n minus k zeros, there are n minus k plus 1 gaps and so among these you need to pick out k gaps.

So, the ones can occupy there are k ones, so the number of choices. So, the number of allowed strings says the number of ways of choosing k gaps from among these n minus k plus 1 gaps. So, what you have is the number h of n comma k is exactly given by n minus k plus 1 choose k. These are the number of ways of picking k gaps out of the total number of caps, and so what is this given as well first we that this problems easier to solve.

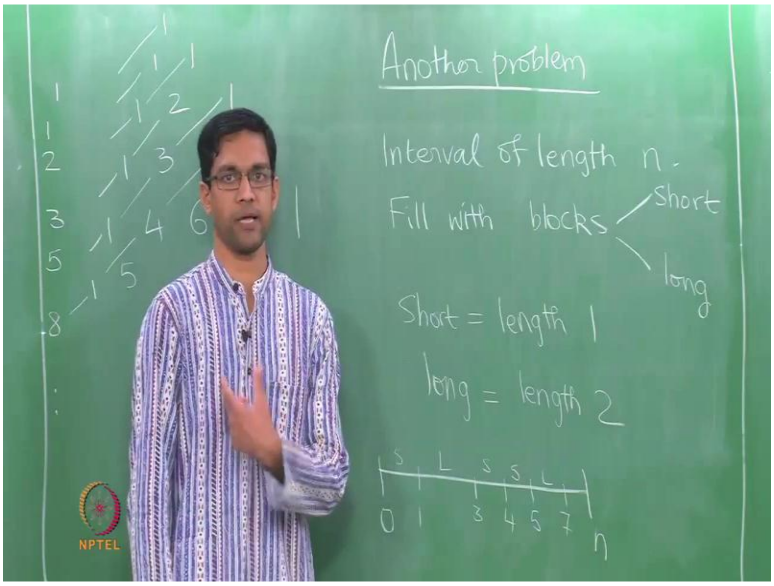
(Refer Slide Time: 08:10)



But, here is another very, very interesting consequence of this, observe that the total number of 0 1 sequences with no successive ones. Well, the total number of each sequence must have some number of ones, it has k ones and what is k, k can be any number between 0 and n. So, in fact, here is the equation which says take a sequence look at how many ones it has, the number of ones has to be something between 0 and n, so that proves that f of n is exactly this sum.

So, what is that mean it says that, so here is our conclusion we just said that this is just the n plus second Fibonacci number. So, from our analysis here is what we have concluded that the n plus second Fibonacci number is exactly certain sum of binomial coefficients.

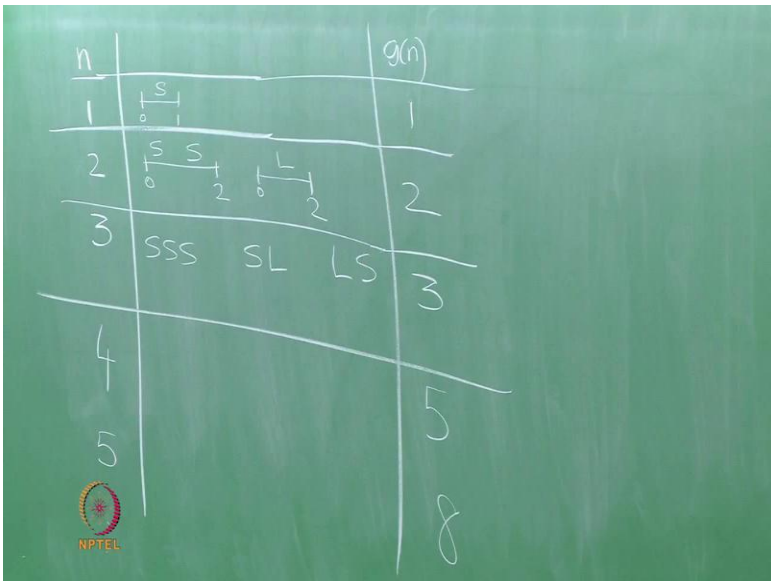
(Refer Slide Time: 09:35)



And so recall that the binomial coefficients are usually arranged in the form of Pascal triangle. So, I have 1 1 1 1 2 1 and so on and so what we have here is just... So, what it says really is the following, if you take the sort of diagonal some. So, I take 1 2, I take... So, if you sort of add up these, so called diagonals what you get or just the Fibonacci numbers. So, this is 1 1 that is a 2, this is a 3 1 3 1 is a 5, 1 4 3 is an 8 and so on, so you can actually obtain the Fibonacci numbers from the Pascal's triangle by summing up along these diagonal in some sense.

So, here is another problem it is says the following you have an interval of length n. So, imagine that you have say between 0 and n, so you have this interval and now what you want is to fill this up with blocks by a block we mean well there are two kinds of blocks there are sort blocks and long blocks, a sort blocks is of length 1. So, imagine that say between 0 and 1 filling like this will be a filling by a sort block between say after a sort block I want to fill it with a longer block, so between 1 and 3 I have a long block. So, I could fill say with two sort blocks and then follow it up with a long block and so on. So, imagine you have to fill up this interval between 0 and n by a succession of short and long blocks, and the question is in how many ways can this be done.

(Refer Slide Time: 12:37)

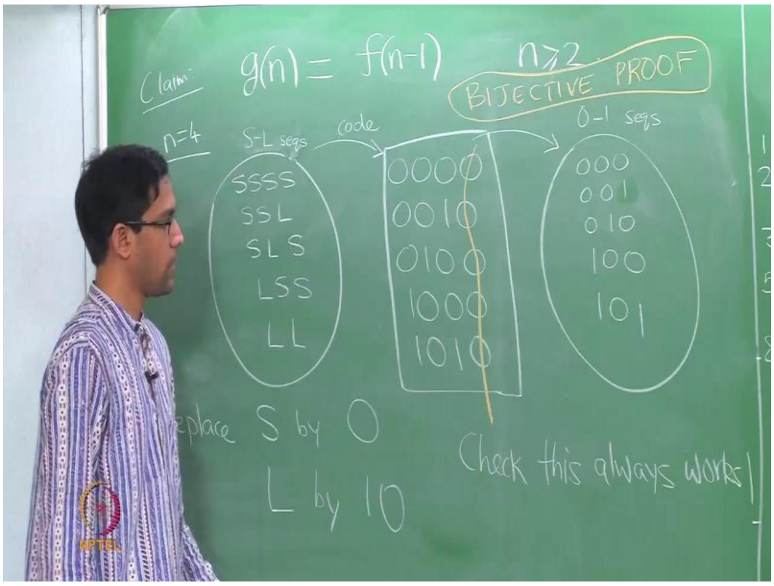


So, let us give this a name let us say the number of ways doing this is let us say g of n. So, of course, a depends on n and so here is the observation we could of course, try this out for some small values of n. So, if you look at a let see n, so if n is 1 what are the possible configurations you can only have a short block, because well the total interval has length 1. So, you can only fill it with the short block, if n is 2 you have an interval of length 2 which you can either fill with two short blocks or with a single long block.

Similar, if n is 3 you can either have 3 short blocks or a short blocks followed by a long block or a long block followed by a short block. So, well what are these numbers the total number g of n. So, in this case 1, in this case 2, in this case it is a 3 and so on, so I am not yet I am not really going to write out the remaining values. Now, here is the interesting observations, so if you do this. So, if you sort of look at what happens for the nest few values, here is what you will find that it actually gives you back the same Fibonacci sequence, so 1, 2, 3, 5, 8 and so on.

So, here again there is an appearance of the Fibonacci sequence and what we like to do well to try and solve this problem, you could try and proceed the same way, try and show that it has the same recurrence relation and so and so forth. But, I want to use this to demonstrate certain form of proof called the proof by bijections. So, here is the claim that we will now prove that this number g of n is the same as.

(Refer Slide Time: 14:46)



So, here is claim that is proof the g of n is actually the same as let us call it f of n minus 1, this is at least n is 2. Remember, f counts the number of 0 1 word's strings of length n with zeros and ones which do not have two successive ones. So, let us proof that the number of you know short and long fillings is, in fact the same as number of 0 1 strings and I will sort of do this by example.

So, let us take the case n equals 4 for which we have the following short and long strings and we will sort with n equals 3 imagine. So, let see n equals 4 has 5, so let us to n equals 4, so I have total interval of length 4, so I have 4 sorts strings may be have 2 shorts and a long string I have a short long short I have long short or I have two longs, so these are the all the ways in which I can, make up total length 4 using just short and long strings.

Now, this number the number of such configurations is exactly counted by... So, I am going to n equals 4, so the number of allowed configurations of shorts and longs is counted by g of n, g of 4 is 5 here on the other side I am going to write out 0 1 sequences. So, what I want is the following I am going to write out 0 1 sequences and these are 0 1 sequences. So, these are short long sequences and on the other side I am going to write 0 1 sequences.

And, but then I am going to write it for n equals 3, so what are the allowed 0 1 sequences for n equals 3, well that is all three zeros 0 1 0 it is 0 0 1, 0 1 0, 1 0 0, 1 0 1. So, there are 5 of them and this is really what I am going to prove that the number of sort long sequences is a same as the number of 0 1 sequences with one smaller length. So, how do we do this, so what is the probe by bijection, what you need to do is to give a recipe which to each of these sequences should map something in the other set and a recipe we should work all values of n, not just for some specific values.

So, here is the sort of here is the algorithm, so let us do this by going via intermediately. So, I will define the following map to each sequence of short and long let us do the following, wherever you see a short. So, replace s by a 0 and wherever you see a long sequence you replace it by 1 0. So, think of it as a code wherever there is a short you replace it just by a 0 and a long by 1 0.

So, if I do this what do I get, so here is my code function, so short becomes all shorts or all zeros, short short long, sort sort long, sort long sort, long means 1 0 short short and two longs is 1 0 1 0. So, what I have done is why are this encoding function I converted this strings of shorts and longs into a string of zeros and ones, but observe I have a string which is of length 4 now. So, that is the resign I said this is a intermediary step what I now have is strings of length 4.

But, here is the key observation all of these strings observe we will always have to end with a 0. I am not going to get all possible strings here, because all them are four times with the 0 and why is that, because if I see a short I am going to put a 0 if I see a short to the end I will put a 0 and if I put a see a long I am going to put a 1 0 any way. So, what I am going to get is always a 0 at the end. So, what I do is I just delete that 0 after all I know all of them are zeros.

So, let me get read of all those zeros, so here is the recipe you first encode this via zeros and ones in that manner and then you get read of all the zeros at the end by doing this what you get a exactly a map here. So, I am going to send s s s s to three zeros s s l will go to 0 0 1 and so on and so forth. So, what I have is a actually a bijection between these two sets a map which is one to one corresponding each thing here maps to single thing there and everything there comes from is map to by something from here.

And so check that this in fact, works in general, so check this always works no at a what my ends. So, this methodology of proof is often what is called bijective proof, so these are often very difficult to construct. So, nice bijections or rather difficult to constructive in general, but when the exist they often give very eliminating proofs of why two numbers must actually be equal.