Discrete Mathematics
Prof. Ashish Choudhry
IIIT, Bangalore

Module No # 05
Lecture No # 37
Tutorial 6: Part 1

Hello everyone, welcome to the first part of tutorial 6.

(Refer Slide Time: 00:28)

In question number 1, we are supposed to find the required relation for the number of strictly increasing sequences of positive integers that has 1 as its first term and n as its last term where n is some positive integer. So for instance, if $n = 1$ then there is only 1 sequence which starts with 1 and ends with 1 and which is a strictly increasing sequence. Whereas as if $n = 2$ then also I have only 1 sequence starting with 1 and ending with 2 and which is strictly increasing.

By strictly increasing sequence, I do not mean that I need all the numbers between 1 to n. Similarly, if $n = 3$ then a possible sequence could be starting with 1 ending with 3. This is a strictly increasing sequence or I can have a sequence where I have $(1, 2, 3)$ starting with 1 and ending with 3 and having 2 in between. So these are the various sequences possible for different values of n. Now our goal is to find out a recurrence equation for this problem.

Basically, what we are asking here is the following. Imagine you have an arbitrary sequence starting with a_1 ending with a_k. The restrictions are here that the first term should be 1, namely

a_1 should be 1 and the last term a_k should be n and in between you have the numbers $a_2 \dots a_{k-1}$ where they appear in the strictly increasing order. They can be any numbers from $\{2 \ \dots \ n-1\}$. So let s_n denotes my functions which is the number of valid sequences ending with n.

So a trivial recurrence condition for s_n is the following. I can say that $s_n = s_1 + s_2 \dots s_{n-1}$. This is because, the second last value in the sequence, namely a_{k-1}, can be 1. If that is the case then basically what I am saying is find out all possible sequences starting and ending with 1 and append it with the value n. That will give you one category of valid sequences starting with 1 and ending with n.

How many such sequences you can have? You can have s_1 such sequences. Or a_{k-1} can be 2. In that case what I am saying is you find out all valid sequences starting with 1, ending with 2, and which is a strictly increasing sequence and in each such sequence you put an n at the end and that will give you another category of strictly increasing sequences starting with 1 and ending with n.

How many such sequences you can have? Can have s_2 number of such sequences and the same way your second last value in the sequence a_{k-1} could be $n - 1$. So basically I am saying take any valid or strictly increasing sequences starting and starting with 1 ending with $n - 1$ and put an n at the end and that will give you another category of strictly increasing sequences starting with 1 and ending with n.

And you can have s_{n-1} such sequences. And all this category of sequences that we have discussed, they are disjoint. So this is one of the recurrence conditions for our function s_n. But we want a more compact recurrence condition. Compact in the sense, the degree of this equation that we had just formulated, what is its degree? Its degree is $n - 1$ because the n-th term of the sequence that we had formulated here depends on $n - 1$ previous values in the sequence.

We want a more compact recurrence condition. So let us derive that alternate recurrence equation and this alternate recurrence equation is derived by using the following argument. So imagine you have, considering a valid or strictly increasing sequence starting with 1, ending with n, and the second last value in the sequence is a_{k-1}. Now there can be 2 categories. Category 1 that your,

a_{k-1} is $n - 1$. That means you are considering all such strictly increasing sequences where the second last term is $n - 1$.

How many such sequences you can have? You can have s_{n-1} sequences and you take any such sequence and append it by a value of $a_k = n$, you get a strictly increasing sequence starting with 1 and ending with n. And there will be s_{n-1} such sequences. Category 2 is the following. The second last value namely a_{k-1} in your sequence can be either 1 or 2 or $n - 2$. And you end this a_{k-1}; you append the n at the end of this a_{k-1} you get a valid or strictly increasing sequence starting with 1 ending with n.

Now what we can do is? We can interpret the n which is there at the last position, last position means after a_{k-1} in your actual sequence ending with n. We can interrupt this last n as $n - 1$. It is just some mental interpretation. So for instance what I am saying is here is one of the strictly increasing sequences in category 2. Why? Because my second last value namely a_{k-1} here, is $n - 3$. It is not $n - 1$ it is $n - 3$ and then there is an n following this $n - 3$.

So I can just view this sequence as another sequence where instead of n being appearing at the last position I have $n - 1$ appearing at the last position. Mind it, I stress here that these second category of sequences are different from the first category of sequence. Because in the first category of sequence I have only $n - 1$ allowed in the second last position. But in this category 2 of sequences I am not allowing $n - 1$ in my second last position.

My second last position could be anything from the set 1, 2 or $n - 2$ and then the last position is occupied by n. What I am thus basically saying is that just read that instead of being n having at the last position, you have $n - 1$ at the last position. And it is easy to see that how many such sequences you will have? By doing this logical interpretation, that means, by substituting the last n with an $n - 1$ in your mind you can easily see that there will be s_{n-1} such sequences.

And also this category 1 of sequences and category 2 of sequences are disjoint. And you cannot have any third category of strictly increasing sequences because your second last position namely a_{k-1} could be either $n - 1$ or different from $n - 1$. And these are the only 2 possible cases which

we have considered and hence we can say that our alternate recurrence equation will be $s_n = 2 * s_{n-1}$.

Now this is an equation of degree 1 because now the dependency of the n-th term is only on the previous term. So definitely this is a better recurrence equation in terms of solving it and in terms of the number of initial conditions that we need here. It turns out that even though this equation; this alternate recurrence equation is of degree 1, we need 2 initial conditions. So $s_1 = 1$ this is because if $n = 1$ there is only 1 sequence which is strictly increasing starting with 1 and ending with 1; namely with sequence 1 i.e., the sequence which is having only the value 1 in it.

And if I take $n = 2$ then also I have only 1 sequence which is strictly increasing; starting with 1 and ending with 2. The sequence will be 1 followed by 2. That is why $s_2 = 1$. Now you might be wondering that why s_2 is explicitly specified as an initial condition. Why cannot I just have an initial condition $s_1 = 1$ because this is an equation of degree 1 here.

This is because if I do not explicitly specify the value of $s_2 = 1$ and then if you try to derive s_2 by substituting $n = 2$ in this recurrence equation you should get $s_2 = 2s_1 = 2 * 1 = 2$, which is not the case because you do not have 2 strictly increasing sequences starting with 1 and ending with 2. You have only one strictly increasing sequence. Or does that mean that our recurrence condition that we have derived is wrong?

Well that is not the case because if you see here the problem that is happening here is that if I take $n = 2$ then this case 2 or category 2 of strictly increasing sequences would not be there at the first place. Because for $n = 2$ this case 2 boils down to saying that at the second last position you have one at the beginning and you're a_{k-1}, namely the second last string, could be from the set 1 to 0 which is not the case.

Because $n - 2 = 0$ for $n = 2$, you cannot have a second last position occupied by something, say 0. That is not allowed. So that is why case 2 or category 2 is not going to occur for the case of $n = 2$. For $n = 2$ it is only the case 1 which can occur namely your second last position can be occupied only with 1. And then you append it with 1 that is why you have only 1 valid sequence for the case of $n = 2$ as well.

It is only from $n = 3$ onwards that your category 2 of strictly increasing sequences appear. And that is why we have 2 initial conditions needed.

(Refer Slide Time 12:50)
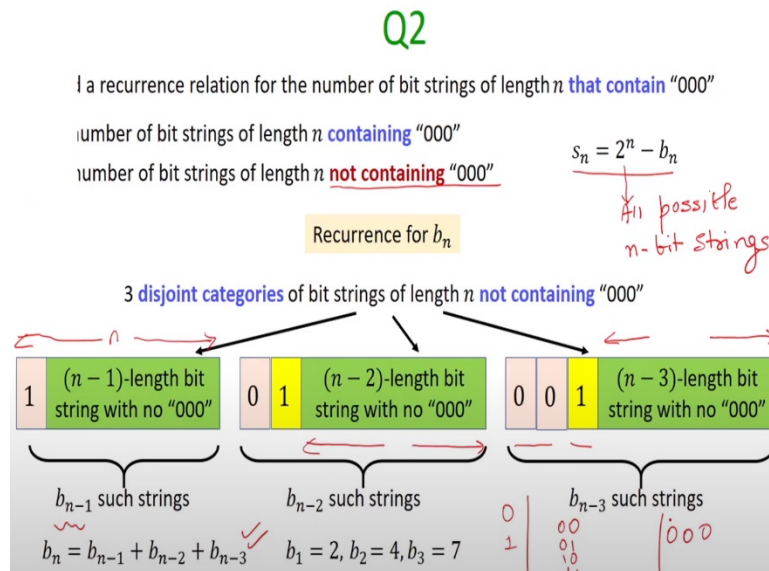
In question 2 you have to find out the recurrence relation for the number of bit strings of length n that contain the substring "000". It might have more than 1 occurrence of "000" as well. In fact, a string of length n consisting of all 0's is a valid string. So let us try to find out this by formulating a recurrence equation. So let s_n denote the number of bit strings of length n containing the substring "000".

So what we will do is instead of counting the number of strings containing "000" we will rather count the number of strings of length n not containing "000" because it will be relatively easier to formulate a recurrence equation for the number of bad string. Here, the term bad string denotes strings that do not have an occurrence of "000". So, I call or denote the sequence of all n length bit strings not containing "000" by this b sequence and b_n is the n-th term of such sequence.

Now it is easy to see that as per the definition of s_n and b_n that $s_n = 2^n - b_n$. This is because 2^n is all possible n bit strings. 2^n denotes all possible n bit strings which has an occurrence of "000" and which do not have an occurrence of "000". And you subtract from such strings all such strings which do not have an occurrence of "000" we will get the number of strings of length n which has an occurrence of "000".

So our goal now will be boiling down to find out the recurrence condition for the b sequence. And it turns out that there are 3 disjoint categories of bit strings of length n not containing "000". Let us look into those 3 categories. Category 1 consists of those strings of length n that start with 1. If it starts with 1 then in order that the overall string does not have any occurrence of "000", it should be the case that the remaining portion, namely the remaining portion of length $n - 1$ bit, should not have any occurrence of "000".

How many such strings we can have? As per our definition of b function, there will be b_{n-1} such strings. So this is category 1 of bad strings. Category 2 of bad strings are those that start with "01" and if that is the case then the remaining $n - 2$ bit positions should not have any occurrence of "000". How many such bit strings we can have? We can have b_{n-2} such strings.

And category 3 of bad strings where the first 2 positions are 0, and in that case, the third position cannot be 0 because if the third position has a 0 then it is not a bad string. The definition of bad string is that it should not have any occurrence of "000". So in category 3 we are considering the case where the first 2 positions are 0 the third position is 1. And after that in order that the overall string do not have any occurrence of triple 0 the remaining substring of length $n - 3$ should not have any occurrence of "000".

How many such bad strings I can have? I can have b_{n-3} such strings. And these are the only 3 categories of bad strings. I cannot have any other fourth category and it is easy to see that they are disjoint. So how many bad strings I can have? The number of bad strings of length n will be the summation of the number of bad strings of length $n - 1$, $n - 2$ and $n - 3$ and hence we get this recurrence equation for the b series.

And since this is an equation of degree 3, we need 3 initial conditions. $b_1 = 2$ because both the string 0 as well as string 1 do not have any occurrence of "000". Similarly, $b_2 = 4$ and $b_3 = 7$.
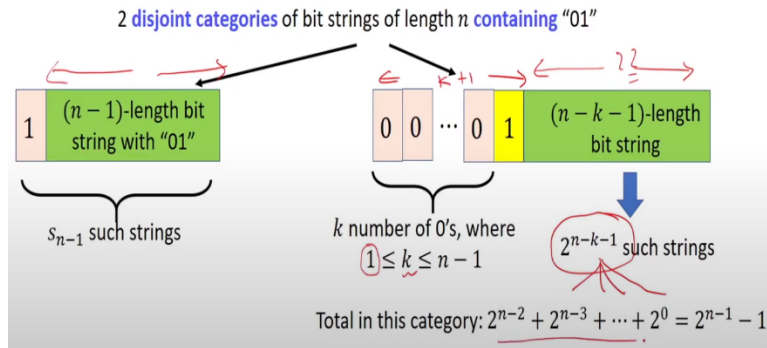**(Refer Slide Time 18:12)**

Q3

Find a recurrence relation for the number of bit strings of length n that contain "01"

$s_n \overset{\text{def}}{=}$ number of bit strings of length n containing "01"

$$s_n = s_{n-1} + 2^{n-1} - 1$$

2 disjoint categories of bit strings of length n containing "01"

Total in this category: $2^{n-2} + 2^{n-3} + \cdots + 2^0 = 2^{n-1} - 1$

So you can see in this question that why we did not formulated directly the recurrence condition for s_n. Because that might be slightly involved but formulating a recurrence of b_n is very easy and that is why we formulated our recurrence condition or expressed our recurrence condition for s_n in terms of b_n.

In question 3, we have to find out the recurrence relation for the number of bit strings of length n that contain "01". So let s_n denote the number of bit strings of length n containing "01". Again, we need at least 1 occurrence of "01" there could be more occurrences of "01" allowed as well. My claim is that s_n satisfies the recurrence condition $s_n = s_{n-1} + 2^{n-1} - 1$. Let's see why. So it turns out that we can have 2 disjoint categories of n length bit string containing "01". Category 1 where the string starts with 1. If it starts with 1 and in order that the overall string has an occurrence of "01" we need that remaining $n - 1$ length bit string should have an occurrence of "01".

As per our definition of s function I will be having s_{n-1} such bit strings. And category 2 where the string starts with 0. Now the strings starts with 0; I may have only 1 zero, or I may have 2 zeros, or I may have a sequence of k zeros followed by a 1 and then I do not care what is the remaining $n - k - 1$ substring. And here k ranges from 1 to $n - 1$. Why from 1 to $n - 1$? Because we definitely need 1 occurrence of 0 followed by a 1.
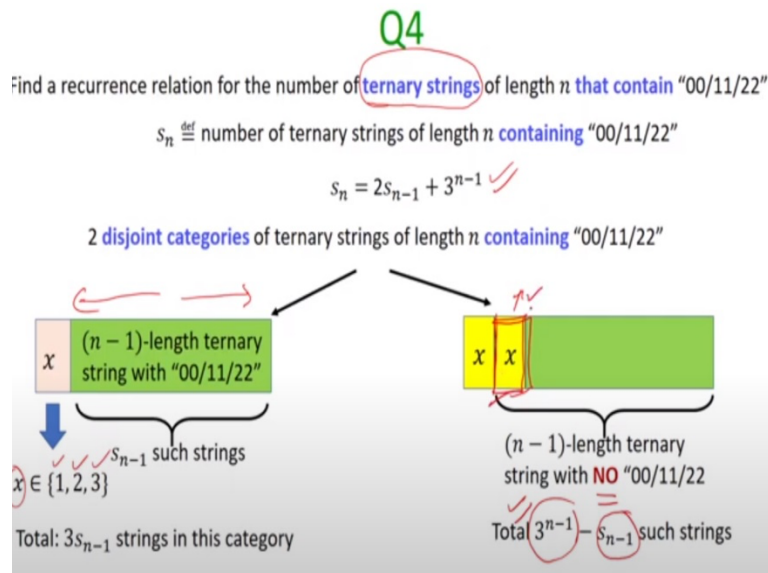
So that is why k = 1. So k denotes the number of zeros which are there at the beginning of the string. So I can have either 1 number of 0, or 2 number of 0, or $n - 1$ number of zeros because I

might be having a string of the form "000...0" $n - 1$ number of time and the last bit is 1. That is also a valid string in this category 2. So that is why my k ranges from 1 to $n - 1$.

And then I do not put any restriction. Once I have occupied $k + 1$ bits here, the remaining $n - k - 1$ positions can be occupied by any bit, either 0 or 1. So how many strings in this second category I can have? It depends upon how many slots I am left here. Because the first $k + 1$ slots are reserved for k number of zeros and one single 1.

So the remaining $n - k - 1$ positions can be occupied by either 0 or 1 and hence I can have 2^{n-k-1} many strings in the second category. So if I sum these 2 things I will get the total number of strings but it turns out that in category 2 my k ranges from 1 to $n - 1$. So that is why I have to put $k = 1, k = 2, k = 3 \dots k = n - 1$. And hence I get total $2^{n-2} + 2^{n-3} + \dots + 2^0$ many strings in category 2 and this is nothing but a geometric progression. If I sum them up I get $2^{n-1} - 1$ value and that is how we get $s_n = s_{n-1} + 2^{n-1} - 1$.

(Refer Slide Time 22:20)

Q4

Find a recurrence relation for the number of ternary strings of length n that contain "00/11/22"

$s_n \stackrel{\text{def}}{=}$ number of ternary strings of length n containing "00/11/22"

$$s_n = 2s_{n-1} + 3^{n-1}$$

2 disjoint categories of ternary strings of length n containing "00/11/22"

| x | $(n-1)$-length ternary string with "00/11/22" |

$\checkmark \checkmark \checkmark s_{n-1}$ such strings
$x \in \{1, 2, 3\}$

Total: $3s_{n-1}$ strings in this category

| x | x | |

$(n-1)$-length ternary string with NO "00/11/22"

Total $3^{n-1} - s_{n-1}$ such strings

Now let us go to question number 4. We want to find out a recurrence relation for the number of ternary strings of length n that has an occurrence of "00" or "11" or "22". It may have occurrence of all of them or it may have an occurrence of "00" as well as "11". I need either an occurrence of "00" or an occurrence of "11" or an occurrence of "22". And by ternary strings I mean that the only characters which are allowed in your string are 0, 1 and 2.

So let s_n denote the number of such ternary strings. The claim is that s_n is satisfying the recurrence condition $s_n = 2s_{n-1} + 3^{n-1}$. So let us derive that. It turns out that we can have 2 disjoint categories of ternary strings of length n. Let's see them. Category 1 where I take a ternary string of length $n-1$ which satisfies my condition, namely it has an occurrence of "00" or "11" or "22". If I take any such $n-1$ length ternary string and there will be s_{n-1} such strings and if I put any character at the beginning of such a string; that character x could be either 1 or 2 or 3 it does not matter because I have already encountered an occurrence of "00" or "11" or "22" in the remaining portion of the string. So it does not matter what is my x or it will give me overall a ternary string of length n satisfying my conditions. So how many such strings I can have? So since my x can take 3 possible values; I have $3s_{n-1}$ strings in this category.

Category 2 where I take an $n-1$ length ternary string which has no occurrence of "00" or "11" or "22". How many such strings I can have? The number of such strings will be all possible ternary strings of length $n-1$ minus the number of ternary string length $n-1$ which do have an occurrence of "00" or "11" or "22". So that will be the total number of strings of this form. And let x be the first position of this $n-1$ length ternary string. Right now there is no occurrence of "00" or "11" or "22".

I want to convert the string into a string of length n which has an occurrence of "00" or "11" or "22" and the only way I can do that is I have to put the same character x at the beginning of this string as well. So that will ensure that the overall length of the string becomes n and then will have an occurrence of either "00" or "11" or "22". So how many strings of this category I will have? It will be exactly $3^{n-1} - s_{n-1}$ because I do not have an option of x here.

Once this x is frozen I have to repeat the same x at the first position. So that is why the total number of strength in this category will be $3^{n-1} - s_{n-1}$ and if I sum them, these two things, I get $s_n = 2s_{n-1} + 3^{n-1}$.

(Refer Slide Time: 26:07)

Q5

$T(m, n) \stackrel{\text{def}}{=}$ number of **onto functions** from A to B

Prove: $T(m, n) = n^m - \sum_{k=1}^{n-1} C(n, k) T(m, k)$

❖ **Subtract** the set of **non-onto functions** from the set of **all functions** from A to B

$|A| = m$ $|B| = n$

❑ Number of functions from A to B --- n^m

❑ Number of **non-onto functions** from A to B --- $\sum_{k=1}^{n-1} C(n, k) T(m, k)$

❖ For any **non-onto function** f', the **image set D** is a **proper subset** of B --- $D \subset B$

➤ $C(n, k)$ ways of selecting D, if $|D| = k$ $1 \le k$ $\le n-1$

➤ f' will be a **onto function** from A to D

▪ $T(m, k)$ onto functions from A to D

$|A| = m$ $|B| = n$ $|D| = k$

Let us go to question 5. In question 5 I have defined the function $T(m, n)$ which is as follows. So you have 2 sets. A set A whose cardinality is m and set B whose cardinality is n and $T(m, n)$ denotes here the number of onto functions from the set A to the set B. I have to show that $T(m, n)$ is satisfying the recurrence condition $T(m, n) = n^m - \sum_{k=1}^{n-1} C(n, k) T(m, k)$. So let us prove that. The idea behind the derivation of this recurrence condition is that since a subtraction is involved here and you can see and identify the structure of this n^m.

n^m denotes the number of all possible functions from A to B. It could be either onto or it could be non-onto. So if I subtract all non-onto functions from the number of all possible functions from A to B that will give me the number of onto functions. So it is easy to see that what this recurrence condition is saying is to derive $T(m, n)$ subtract the number of non-onto functions.

That means I have to show that this latter part of this recurrence condition denotes the number of non-onto functions from A to B and indeed that is the case. So why that is the case? So you are given the sets A and B and we want to find out the number of non-onto functions. What will be the characteristic of a non-onto functions? A function will be a non-onto function from A to B if there is at least 1 element from the set B which do not have any pre-image.

Or in other words I can view any non-onto function as follows. So let f' be an arbitrary non-onto function from A to B. If I focus on the set of possible images which are assigned as per this function

f' and call that set as D, the image set i.e., D is the set of all the elements from B which are the images as per the function f'.

Then since my function f' is a non-onto function clearly this image set D will be a proper subset of B. So more specifically if the cardinality of image set D is k; $k < n$. k could be either 1, k could be either 2, k could be either 3 or k could be either $n - 1$. If you ensure that if I take these values of D then it ensures that there is at least 1 value from the set of B which is left which is not going to have any pre-image.

So now in how many ways I can select my set D? Since its cardinality is k, I can select it in $C(n, k)$ ways and as I said k ranges from 1 to $n - 1$. So that takes care of the fact that I have to consider a summation from $k = 1$ to $n - 1$. Now once I have decided the value of k, I have to find out which subset or what should be my subset D of size k. So I have n choose k or $C(n, k)$ ways of selecting my set D of size k.

And once I have chosen by set D, the set or the function f' will be an onto function from the set A to D. That means I if I just forget about the set B and just imagine that I have a set D of size k and consider an onto function from the set A to the set D that can be interpreted as a non-onto function from the set A to the set B because my set D is a proper subset of the set B. So now how many such onto functions I can have from A to D?

So as per my definition of the T function, there are $T(m, k)$ such functions. So once I have decided the cardinality of the set D, I have $C(n, k)$ ways of selecting that as per the product rule. For each way of selecting a subset D of size k, I have $T(m, k)$ number of onto functions from the set A and the chosen D set and hence if I multiply that, that will give me the number of non-onto functions where the image set of those non-onto functions is the set D.
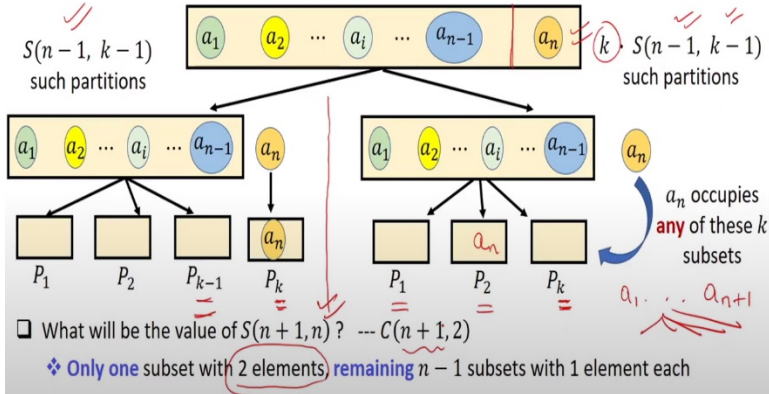
And then as I said if I take the summation from $k = 1$ to $n - 1$ that will give me the overall number of non-onto functions.

(Refer Slide Time: 31:41)

Q6

$S(n,k) \stackrel{\text{def}}{=}$ number of **partitions** of an n-element set into k **non-empty disjoint subsets**

Prove: $S(n,k) = S(n-1, k-1) + k\, S(n-1, k)$

❑ **Two disjoint categories** of partitions of any set $X = \{a_1, \cdots, a_n\}$ into k disjoint subsets

$S(n-1, k-1)$ such partitions

$k \cdot S(n-1, k-1)$ such partitions

a_n occupies **any** of these k subsets

$a_1 \ldots a_{n+1}$

❑ What will be the value of $S(n+1, n)$? --- $C(n+1, 2)$

❖ **Only one** subset with 2 elements, remaining $n-1$ subsets with 1 element each

Let us go the question 6. Here we recall our sterling functions of type 2, namely the $S(n,k)$ function which is the number of partitions of an n element set into k non-empty disjoint subsets and we have to prove that it satisfies this recurrence condition. So, since the plus is involved here definitely I have to show that I have 2 different categories or disjoint, categories of partitions. Indeed that, is the case.

So consider an arbitrary set X consisting of n elements which I denote as a_1 to a_n. And I have to find out in how many ways I can partition this collection into k disjoint subsets. Such that if I take the union of those subset I get the whole set X and the subset should be pairwise disjoint. So I can have 2 categories of partitions. Category 1 where I divide or I partition the first $n-1$ elements into $k-1$ subsets; that means I am not considering a_n right now.

I am imagining that I have only $n-1$ elements and I want to partition them into $k-1$ subsets. And then create an extra subset which only has the element a_n. That will now result in total k number of disjoint subsets which will whose union will cover the entire set X. So how many such partitions I can have? The number of such partitions is nothing but the number of partitions possible for partitioning the $n-1$ elements into $k-1$ disjoint subsets and as per the definition of S function $S(n-1, k-1)$ is the number of such partitions.

Category 2 where I take the $n-1$ elements and partition it into k disjoint subsets not $k-1$. Now, I have to still put a_n somewhere. In this category, what I am going to do is I am going to consider

the case that a_n is going to occupy any of these k subsets. So this is clearly different from the case where we have explicitly reserved a whole partition P_k for the n-th element. And the remaining $k - 1$ subsets where covering the remaining $n - 1$ elements.

Here the entire k subsets are going to cover the $n - 1$ elements and a_n also is supposed to occupy 1 of those subsets. So it is easy to see that the number of partitions in this category will be first we have to find out how many ways I can partition $n - 1$ elements into $k - 1$ subsets. And once I have decided the subset P_1 to P_k, I have the option of either putting a_n namely the n-th element in P_1 that is 1 option, or I have the option of putting a_n into the second subset and so on.

And that is why k is multiplied here because of the product rule. And if I sum these 2 categories of partitions that will give me the overall number of partitions. So what will be the value of $S(n + 1, n)$? The number of partition of an $n + 1$ element set into n non-empty disjoint subsets. It is nothing but $C(n + 1,2)$ because if I have $n + 1$ elements namely a_1 to a_{n+1}.

And if I want to divide it into n subsets then that means that there will be only 1 subset with 2 elements and remaining all other subsets will have 1 element each. So now I have to just find out the 2 special elements which are going to occupy that subset which is allowed to have 2 elements and that will automatically ensure that the remaining $n - 1$ elements which are left each of them go to a single distinct subset. So that is why the answer here is $C(n + 1,2)$ because there are $C(n + 1,2)$ ways of picking 2 elements from $n + 1$ elements.

So it could be either a_1, a 2 which go together in one subset. And remaining all elements go to single distinct subset each or it could be a_1, a_3 which go together in 1 subset and remaining elements go to single distinct subsets each and so on. That is why the answer is $C(n + 1,2)$.

(Refer Slide Time: 37:09)

$$\sum_{k=1}^{n} k \cdot \binom{n}{k}^2 = n \cdot \binom{2n-1}{n-1}$$ Prove using a **combinatorial proof**

❑ **Setting**: n mathematics professors and n computer science professors

❑ **Goal**: to select a committee of n professors whose head is a mathematics professor

LHS	RHS
❖ Let the committee has k mathematics professors and $(n-k)$ computer science professors	❖ First select the **committee head**
$1 \leq k \leq n$	➤ Can be done in n ways as there are n mathematics professor
❖ $\binom{n}{k} \cdot \binom{n}{n-k}$ ways of selection $\approx \binom{n}{k}^2$	❖ Select the **remaining** $n-1$ committee members from the remaining $2n-1$ professors
❖ For each of the above selections, the **committee head** can be any of k mathematics professors	➤ Can be done in $\binom{2n-1}{n-1}$ ways

In question 7, we have to give a proof for combinatorial proof for this identity. So let's see the following setting. You are given n mathematics professor and n computer science professor; you can imagine like that and say that your goal is to select the committee consisting of total n professors where the head of the committee is a mathematics professor. That is my goal. Now I will show that the number of committees satisfying this goal is the expression both in your LHS as well as in the RHS.

So why LHS expression satisfies or gives a number of committees satisfying my goal? Because suppose the committee has k mathematics professors and k will range from 1 to n. Definitely I have to include 1 mathematics professor because the head will be a mathematics professor and in the worst case I can have all the committee members being the mathematics professors. In that case the remaining $n - k$ committee members will be from computer science.

So that tells you that how many ways you can pick the committee. You first pick the k mathematics professors; that automatically ensures that the remaining $n - k$ committee members are from computer science. So they can be selected in $C(n, n - k) = C(n, k)$ many ways. And once you had decided which k mathematics professors are going to be there and which $n - k$ computer science professor are there the committee head itself can be selected out of those k chosen mathematics professors in k ways.

It could be either the first chosen mathematics professor or the second chosen mathematics professor or the k-th chosen mathematics professor. So that is why you multiply it with k because of the product rule and that tells you that why the LHS expression gives you the number of committee members satisfying my goal. Let us see the RHS expression. I can say that in order to satisfy my goal I have to do the following. I have to first select the committee head and this can be done in n ways because there are n mathematics professors.

So that is why this n here. Now once I have decided the committee head I have to still choose the remaining $n - 1$ committee members. And they can be chosen from n computer science professors and the remaining $n - 1$ mathematics professors. Namely I have still $2n - 1$ professors left. I have to choose $n - 1$ members out of them which can be done in $C(2n - 1, n - 1)$ many ways. And now by the product rule, if I multiply the number of ways in which I can select the committee head and the number of ways I can pick the remaining committee members I get the number of committees satisfying them. So with that we end our part 1 of tutorial 6. Thank you.