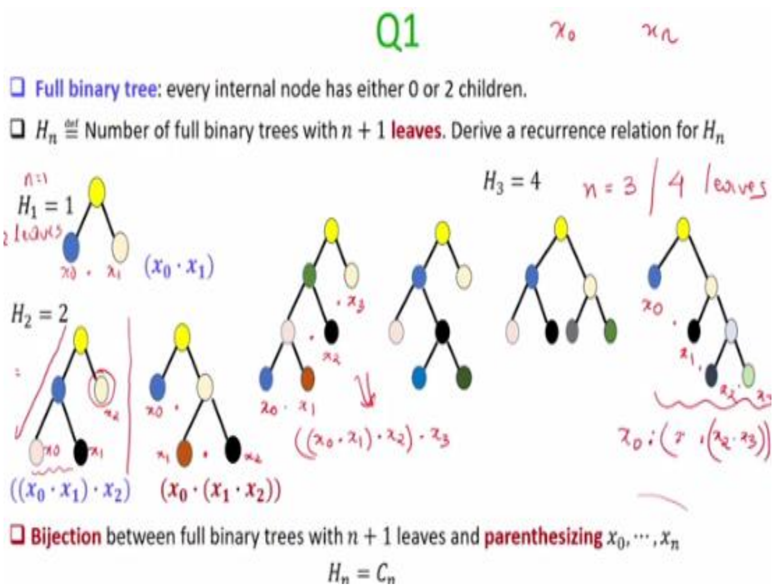
Discrete Mathematics
Prof. Ashish Choudhury
Department of Mathematics and Statistics
International Institute of Information Technology, Bangalore

Lecture -43
Tutorial 7

Hello everyone welcome to tutorial number 7, so let us start with question number 1.

(Refer Slide Time: 00:26)



So, we define first what we call as full binary tree and it is a binary tree where every internal node and by internal node I mean the nodes which are non-leaf nodes, so every internal node has either 0 child or 2 child. And H_n is defined to be the number of full binary trees which has n + 1 leaves. So, we have to derive a recurrence relation for this H_n, so let me first demonstrate the value of H_n for some small values of n.

So, if I consider H_1 that means I want to find out the number of full binary trees which has 2 leaves and there is only one full binary tree which has 2 leaves namely this binary tree. By the way here I am not interested in the label of the nodes I am interested in only the structure of the nodes, so there is only one possible structure of a full binary tree possible which has 2 leaves. What will be H_2?

So, there are 2 structurally different full binary trees which as 3 leaves this is binary tree number 1 this is binary tree number 2. If I consider H_3, H_3 basically denotes a number of structurally different full binary trees with 4 leaves. So, this is one of the trees this is second tree, this is the third tree and this is the fourth tree. As I said earlier I am not focusing on the label of the node.

So, you cannot say that a tree where this node root has label a_1 and the leaves have label a_2, a_3 is different from a tree where the label of the root is a_2 and the leaves are a_1 and a_3. No I am not focusing on the label of the nodes and just focusing on the structure of the tree that is all. So, my claim here is that, the number of full binary trees with n + 1 leaves is same as the value of nth Catalan number and for that what we can do is the following.

Either you can try to derive a recurrence relation explicitly for H_n but we will not do that instead what we will do is we will say that; we will establish a bijection between the set of full binary trees with n + 1 leaves nodes and a set of all ways of parenthesizing n + 1 values to specify their multiplication order and we know already that the number of ways of parenthesizing or the cardinality of the number of ways of parenthesizing n + 1 values is nothing but the nth Catalan number, we already know that.

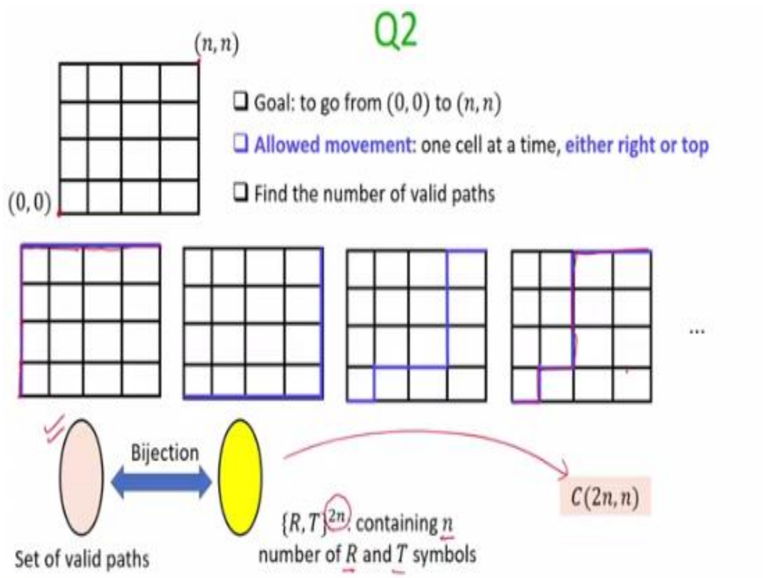
So, if we establish a bijection between the set of solutions for this new problem and the existing problem whose solution we know is Catalan number then we are done. The bijection is easy to formulate. I am not giving the exact details but just giving you a high level idea. So, I have to do the following: I have to take each and every full binary tree with n + 1 leaves. And corresponding to that I have to specify what exactly is the parenthesizing or multiplication order that I can formulate for n + 1 leaves.

So, the n + 1 leaves, I call them as x_0 to x_n, so in this case my n is equal to 1 so that is why I am taking the case of 2 leaves, my 2 leaves are x_0 and x_1 and the only possible parenthesizing here is that I want to multiply them because I have a kind of a balanced tree here. Whereas if I consider this tree for the case where n is equal to 2. So, this is one possible multiplication ordering and this is another possible multiplication ordering.

The multiplication ordering corresponding to these 2 trees are respectively this. So, you can interpret here that this tree is left indented that I am going down first left, so that is why x_0 is multiplied with x_1 first and then whatever is the result that is multiplied with x_2, so treat it as follows x_0, x_1, x_2. Whereas the next tree that you have you have x_0, x_1 and x_2. So, it is equivalent to saying that x_0 is going to be multiplied with the product of x_1 and x_2.

If I take the case of n equal to 3 that means 4 leaves, then this ordering corresponds to $x_0 . x_1 . x_2$ and then multiplied with x_3. So, that will be the order: x_0 getting multiplied with x_1 then that result getting multiplied with x_2 and that result getting multiplied with x_3. Whereas if I consider this tree this will be equivalent to x_0 multiplied with x_1 that is getting multiplied with x_2 and x_3 here. So, it is x_2 and x_3 that gets multiplied then their product is getting multiplied with x_1 and their product is getting multiplied with x_0. So, that is the bijection here.

(Refer Slide Time: 06:28)



In question 2 you are given the following you are given a square grid where you have the coordinate (0, 0) and you want to go to the coordinate (n, n) and you have cells here and the only movements which are allowed to you is that you can at a time you can either go one cell either to the right from the current cell or to the top of the current cell and you have to find out the number of valid paths.

So, for instance: one valid path could be that I go top from the current cell and then again I go top and then again I go top then I go top and then I do right, right, right and right. Whereas I can take a path where I go right and top, right and top, top, top, right and right. Whereas I cannot do the following. I am not allowed to do the following that I go right and right and then again come back left and then top and then bottom.

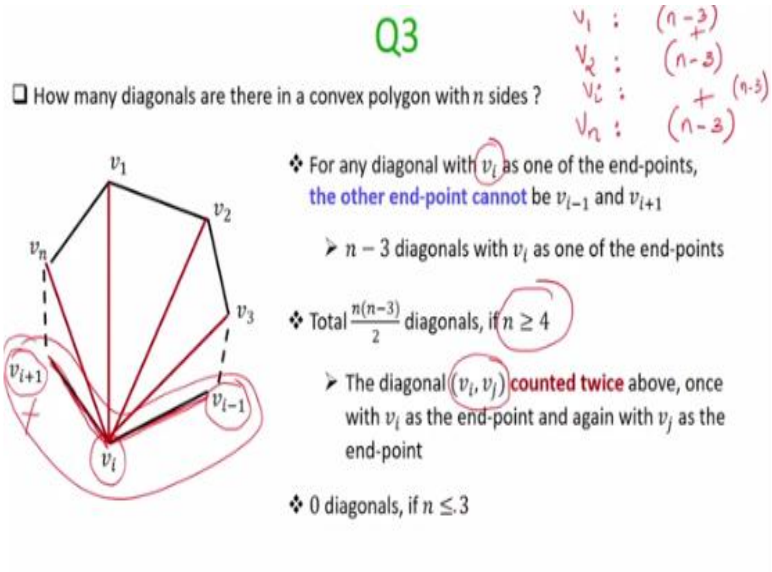
And I cannot do all those things that is not allowed, the only movements that are allowed are one cell at a time either to the right or to the top. So, for solving this or finding the number of valid paths I do the following so imagine this is the set of all your valid paths and you have another set which denote the set of all strings of length 2n. And the 2n length string has equal number of R symbols and T symbols that is a second set, and my goal here is that I will show that there is a bijection between these 2 sets.

And if indeed there is a bijection between these 2 sets then it shows that the number of valid paths is nothing but the number of strings of length 2n which has equal number of R symbols and equal number of T symbols. And we know that the cardinality of this latter set namely the cardinality of the set of all strings of length 2n which has n number of R symbols and n number of T symbols is C(2n, n), because it is equivalent to saying that you have 2n positions.

And out of those 2n positions you have to find out the n positions where the R symbol is going to be there because once you identify the n symbols and positions where the R symbol is going to be there the remaining n positions have to be occupied with the T symbols. And the bijection is very simple here, you take any valid path it will have definitely n number of R movements and n number of T movements because you are at position number (0, 0) and you have to go to the position number (n, n).

The number of R movements; first of all the only movements are the right movements and the T movements, I can imagine that each time I make a right movement I put down a character R. And each time I make a top movement I write down the character T. So, definitely I have to make order n number of R movements and n number of T movements. It cannot be the case that the number of R movements is more than the number of T movements or vice versa because then definitely that is an invalid path and the bijection is very straight forward here.

(Refer Slide Time: 10:35)



The third question is how many diagonals are there in a convex polygon with n sides? So, again we can derive the formula using induction or some other mechanism but we will count it directly. So imagine you are given a convex polygon consisting of n sides and where the vertices are v_1 to v_n, now let us focus on some arbitrary vertex v_i and try to count the number of diagonals that we can have where v_i is one of the end points.

Now if v_i is one of the end points of the diagonal then the other end point of the diagonal cannot be the immediate neighbors of v_i, namely the vertex number v_{i+1} that cannot be the end point because v_i, v_{i+1} is not a diagonal, it is the edge of the convex polygon. In the same way, the other endpoint of the diagonal cannot be v_{i-1}, because v_i and v_{i-1} constitutes an edge or a side for the convex polygon and the side of a convex polygon cannot be treated as a diagonal.

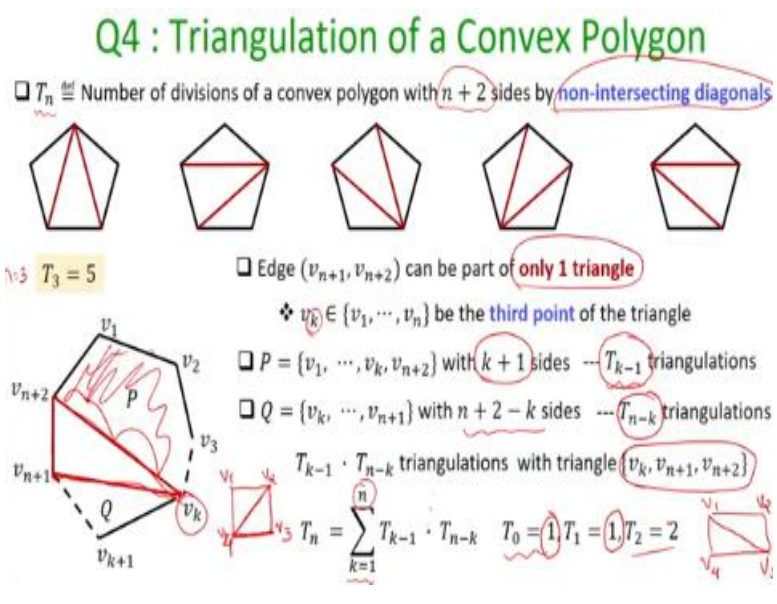
So, that means with v_i as one of the end-points of the diagonal I can have n - 3 diagonals, why n - 3? Because excluding these 3 vertices namely v_i, v_{i-1} and v_{i+1} all the remaining n - 3 end points can be the other end point of the diagonal with v_i being one of the end points. So, now it turns out that the total number of diagonals will be $\frac{n(n-3)}{2}$.

Why over 2? Because what we did here is with v_i being one of the end points I have n - 3 diagonals, so that means with v_1 as one of the end points I can have n - 3 diagonals with v_2 as one of the end points I can have n - 3 diagonals and in the same way with v_n as one of the end points I can have n - 3 diagonals. So, if I sum all of them that gives me the total number of diagonals.

But I will be counting the diagonals twice; I will be counting some of the diagonals twice namely the diagonal with the end points v_i, v_j will be counted twice because with v_i being one of the end points and the other end point could be anything I would have accumulated n - 3 diagonals. One of the diagonals there will be the diagonal where v_j is one of the end points and the same diagonal will be counted again when I will be focusing on the case where v_j is one of the end points and the other end point could be anything including v_i.

So, that is why to compensate or to remove that over counting we are dividing it by 2 and this will be the case where n is greater than equal to 4 because if the number of sides is 3 or less than 3 then we cannot have any diagonal, so for instance in a triangle you do not have any diagonal.

(Refer Slide Time: 13:59)



In question number 4 we are interested to find out the number of triangulations of a convex polygon. So, let T_n denotes the number of triangulations of a convex polygon with n + 2 sides and what is basically a triangulation: it is the process of dividing a convex polygon by non intersecting diagonals. So, if my n is equal to 3 that means I have 5 sides, so I am taking the case of a pentagon and there are 5 ways of triangulating it by non-intersecting diagonals.

So, now I want to find out a recurrence relation or want to find out the number of ways of triangulating a convex polygon with n + 2 sides so I will find that by formulating a recurrence equation and by showing that the solution for that reference equation is same as the nth Catalan number. So, imagine you are given a convex polygon consisting of n + 2 sides which has the vertices v_1 to v_{n+2}.

Now to solve my problem of triangulating a convex polygon with n + 2 sides into smaller problems I consider an arbitrary edge, so for simplicity I take that arbitrary edge to be the edge or the side v_{n+1} and v_{n+2}, namely the side with end points v_{n+1} and v_{n+2} and it turns out that this side v_{n+1}, v_{n+2} can be the part of only one triangle in the overall triangulation. That means if I fix the third point v_k where the v_k could be any vertex in the set v_1 to v_n.

That will fix the triangle in the triangulation which could result in a triangle where one of the sides of the triangle is v_{n+1}, v_{n+2} that means I am focusing up on the case where, so for instance if I could have a case where this v_{n+2}, v_{n+1} being one of the sides of the triangle and the third point would have been v_2. But even in that case this edge v_{n+1}, v_{n+2} can be part of only one triangle; it cannot be part of multiple triangles, because of this reason that we are interested in doing the triangulation using non-intersecting diagonals. So, I will be dividing my problem into smaller problems depending upon the third point or the third vertex of the triangle involving the edge v_{n+1}, v_{n+2}. So, the third vertex v_k could be any vertex it could be the vertex number v_1 in which case my triangle would have been something like this or my vertex v_k could be vertex number v_2 in which case my triangle would have been something like this and so.

So, now once I fix the third vertex namely the vertex v_k the overall polygon with n + 2 sides will be now divided into 2 smaller polygons. The first polygon P which has the vertices this portion this vertex number v_1, v_2,v_3, v_k and the side v_k, v_{n+2} and the last side being v_{n+2}, v_1 so it will have k + 1 sides and how many ways I can triangulate the polygon P? Since it has k + 1 sides as per my definition n + 2 side convex polygon is triangulated in T_n number ways.

So, k + 1 side polygon will be triangulated into T_{k-1} number of ways or T_{k-1} number of triangulations are possible for the polygon P. And the other polygon being the polygon Q, which will have n + 2 - k sides namely the starting vertex will be v_k and the side v_k, v_{k+1} then the next side will be v_{k+1}, v_{k+2} then all the way to the vertex v_{n+1} and the last side being the side v_{n+1}, v_k. And as per my definition of T_n the number of ways of triangulating Q is T_{n-k}.

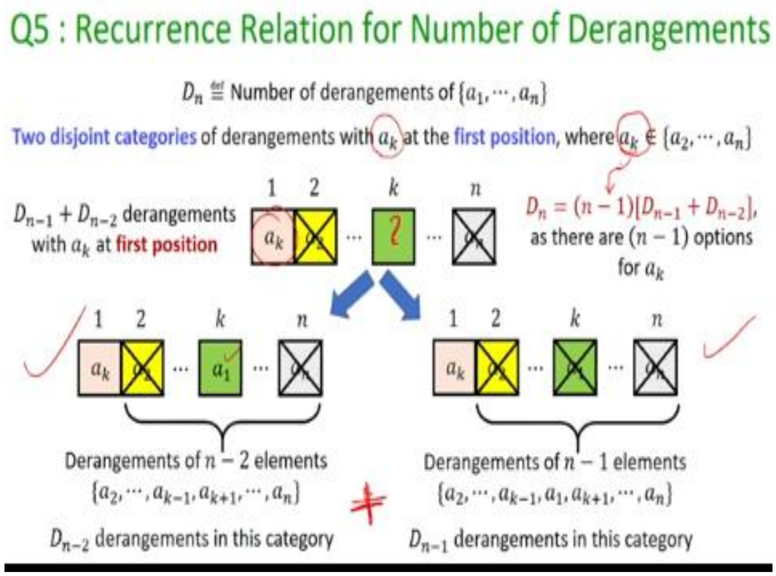
So, these many triangulations for P and these many triangulations for Q along with the triangle where the 3 vertices are v_{n+1}, v_{n+2} and v_k gives me all possible triangulations where one of the triangles is v_{n+1}, v_{n+2} and the third vertex of that triangle is v_k. So, from the product rule it comes out that the total number of triangulations with this being one of the triangles in the triangulations is $T_{k-1} * T_{n-k}$.

But now my k can range from 1 to n, my k could be vertex number v_1, my k could be vertex number v_2 and so on. So, if I take the summation over k being equal to 1 to n then I get the total number of triangulations for n + 2 sided convex polygon and this is the same as the recurrence relation for your nth Catalan number and your initial conditions will be this T_0 is 1, T_0 means the number of triangulations for a 2-sided convex polygon.

Well a 2-sided convex polygon cannot be divided or triangulated so no way of triangulating I am denoting as one way. T_1 means the number of ways of triangulating a convex polygon of 3 sides namely the triangle, and the triangle itself is the triangulation of itself. So, that is why there is only one way. T_2 that means the number of ways of triangulating a rectangle, so this is one possible triangulation and another possible triangulation for the rectangle will be if you draw this.

So, there are 2 ways of triangulating a rectangle that is why T_t will be 2 and from 3 onwards the recurrence will trigger.

(Refer Slide Time: 22:21)



The last question is we want to find out a recurrence relation for the number of derangements of n objects, so just to recap a derangement of n objects is a permutation of those n objects such that none of the objects is at its correct position. That means the object number 1 is not at the first position object number 2 will not be at the second position and so. So, we can divide the set of the derangements of n objects into 2 categories.

So, and for these 2 categories we consider or focus on the element which is there at the first position so we are considering the case where at the first position we have the element a_k where a_k could be either a_2, a_3 or a_n and with a_k being occurring at the first position of the derangement we can have 2 categories of derangements. Category one, where the element a_1 is occurring at the kth position and remember element a_1 is allowed to be occurring at kth position in a valid derangement, because at the kth position we are not putting a_k. a_k has already occupied the first slot and in the kth slot I can have either a_2 or I can have a_3 or I can have a_4. So, I am considering the case where at the kth position the element a_1 is there. So, if that is the case that means I have already deranged 2 objects the kth element is deranged and it is now shifted to the first position and the element a_1 is also deranged it is no longer occurring at the first position but rather occurring at the kth position.

If that is the case then my problem boils down to the problem of the deranging the remaining n - 2 elements so I am still left with the problem of deranging a_2 a_3 a_{k-1} a_{k+1} and a_n. So, I am still left with n - 2 elements and whatever way I could derange them you take any derangements of these n - 2 elements in that the derangement you add the following positioning. You put a_k at the first position and you put a_1 at the kth position that will give you a derangement for the n elements.

And as per my definition of derangement there are $D_{n - 2}$ ways of deranging the n - 2 elements, so this gives you the first category of derangements where a_k is occurring at the first position. The second category of derangements with a_k occurring at the first position is the following. You do not have the element a_1 allowed at the kth position, that means element a_1 can take any other position of course it cannot take the position a_1.

Because at the first position you have already put the element a_k and you are not allowing the element a_1 to occur at the kth position as well in this second category of derangements. So, these are the only 2 categories having fixed the first position or having reserved the first position for element a_k you have only 2 choices or 2 categories of derangements possible. One category where at the kth position you are allowing a_1 and another category of the derangements where you are not allowing a_1 at the kth position.

So, we are considering now the later category, so if a_1 is not allowed to occupy the kth position then I am still left with the problem of deranging n - 1 elements, why n - 1 elements? Because even the element a_1 is now supposed to be kind of deranged in the sense that it is not allowed to be occupying or it is not allowed to occupy the kth position. So, that is like a restriction with respect to element a_1 itself now, so as per my definition of derangements there are $D_{n - 1}$ ways of deranging n - 1 elements.

So, you find out those derangements namely the derangements of n - 1 elements and in that derangement the element a_1 would not be occupying the kth position. You take any such derangement and add the element a_k at the first position that will now give you a derangement of n objects where element a_k is occurring at the first position and element a_1 is not occurring at the kth position and this will be your later category of derangements.

And if you sum these 2 derangements the number of derangements in category 1 and the number of derangements in category 2 that will give you all possible derangements where the element a_k is occurring at the first position. Now since there are n - 1 options for a_k so my a_k could be a_2, my a_k could be a_3, my a_k could be a_n and for each possible a_k I have $D_{n-1} + D_{n-2}$ number of derangements. So, that is why the overall formula for the number of derangements will be n - 1 times the summation of D_{n-1} and D_{n-2}, with that I end tutorial number 7, thank you.