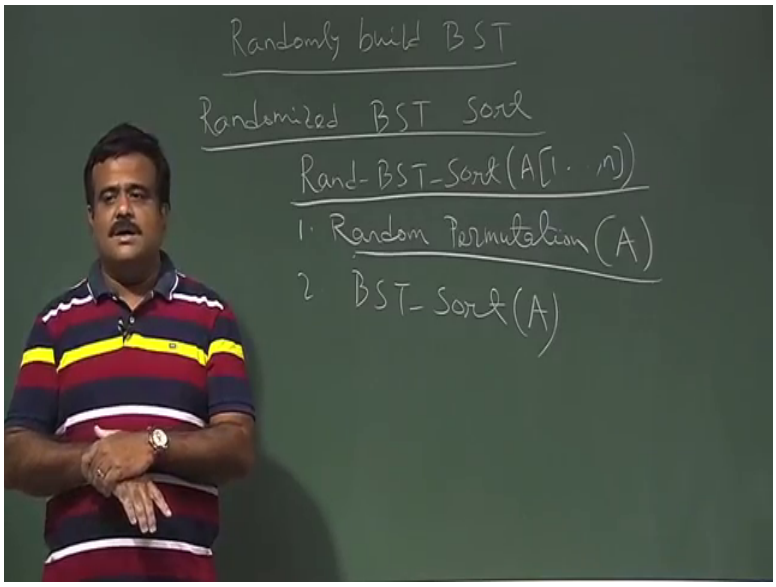
An Introduction to Algorithms
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 26
Ramdomly Build BST

So we will talk about Randomized BST Sort. We have seen the BST sort and we have seen the relationship of BST sort and quick sort they have basically the same number of comparison they are doing. So, the time complexity of BST sort is same as the time complexity of quick sort.

(Refer Slide Time: 00:52)



So, we will talk we will see the idea is to see the randomly built BST sort the; so for that let us look at the randomized version of the BST sort randomized BST sort.

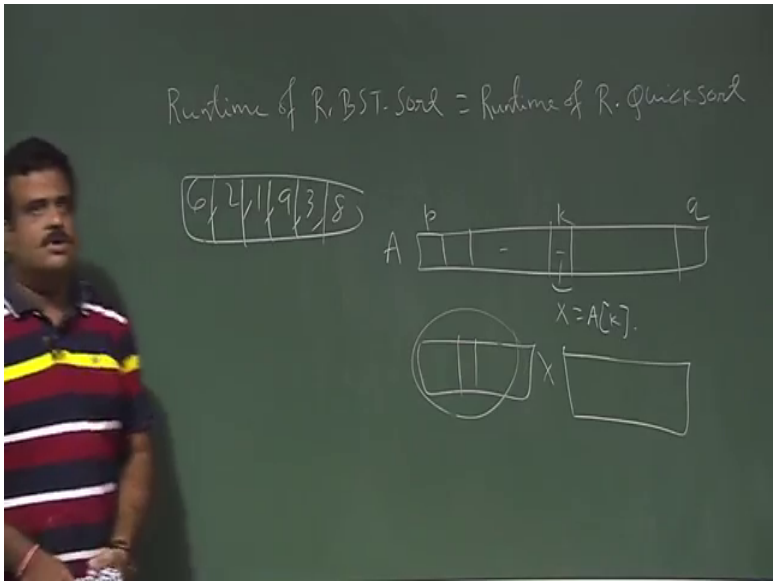
So, for this; what we do? So, this is we can denote by rand BST. So, we have given a n numbers we have given a array of n element we have given n numbers. So, how we can have a sorting algorithm?

So, randomized BST sort means before doing anything we just randomly permute this. So, we will do the random permutation we apply random permutation on this numbers and then call the then we call the BST sort. So, basically we have n number we are given a n numbers and we are just randomly permuting these numbers and then we have

performing the we are building the tree BST; BST and then we are doing the in order traversal. So, this is the randomized version of the BST sort. So, just we are doing the random permutation before we start before we from the BST so, and that BST called randomly build BST. So, given n numbers before building the BST we just have a random permutation on this number linear permutation. So, then we form the BST and then we do the in order traversal and then that will be that will give us a sorting algorithm.

So, this is referred to as randomized BST and this is same as the if we just look at consider the randomized quick sort in the randomized quick sort also what we are doing we are choosing the pivot element randomly and then we are partitioning into 2 parts so; that means, the time complexity of the BST sort we have seen the BST sort and the quick sort having same time complexity. So, the time complexity of the BST sort is same as time complexity of the; is time complexity of randomized.

(Refer Slide Time: 03:32)



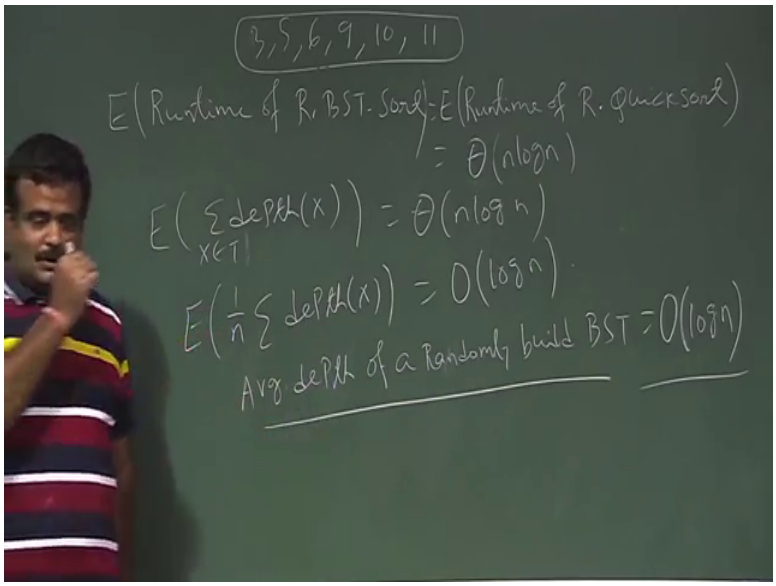
This BST sort is same as time complexity of randomized quick sort. So, this means the time runtime run time of randomized BST sort is same as runtime of randomized quick sort.

Because in randomized quick sort we are choosing the pivot element randomly and then we are performing the partition and that will divide into 2 parts again we are choosing the pivot in the; so it will divide into. So, what is in the randomized fixed we have given

array. So, what we have doing we are choosing the pivot randomly. So, suppose this is our p to q. So, we have choosing a element index A k randomly and this is going to be our pivot and we are partitioning this array into 2 parts x is sitting here like this. So, again we will call randomized quick sort on this randomized quick sort on this. So, again it will be choosing a element randomly from here. So, this is form a tree and this trees same as the randomly build BST because randomly build BST means we have given some numbers we are doing the we have given some numbers . So, this is the number we have given.

Now, we have doing a random permutation on this. So, there are how many permutation there are if there are n numbers there factorial n permutation. So, among these we are choosing one permutation randomly. So, equally likely and then we are forming the tree. So, this is basically some as this randomized version of the quick sort. So, that is why their time complexity is same. So, now, this is a randomized algorithm. So, we need to talk about the average case analysis.

(Refer Slide Time: 05:45)



So, you take the expected value of this. So, expected runtime is same as expected runtime of randomized quick sort now we know this expected runtime of randomized quick sort is order of n log n.

So, now what is the runtime for BST sort runtime of BST sort is basically the time it required to form the tree BST insert. So, this is basically summation of depth of x that x

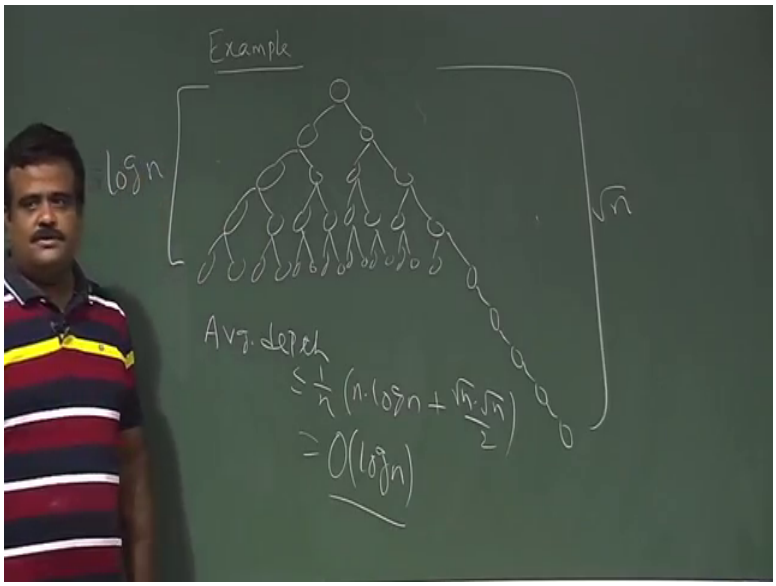
is t. So, this is basically or we can say big o or big theta n log n. So, now, we take the same so, expected. So, this is basically 1 by n. So, this is giving us order of log n. So, this is basically average depth. So, this is basically average depth. So, this is basically average depth expected on average depth of a randomly build BST is basically order of log n.

Actually our goal is to know the expected height of this randomly build BST and that is going to be the; we have to show that is basically the log n; so, expected height. So, expected height of this be a randomly build BST randomly build BST means we just perform a random; random permutation on these numbers and then we form the BST and magic that it will be expected it will be a balanced tree usually the tree is not balanced depending on the input suppose input is say sorted say 3 5 6 9 10 11. So, for this input if we form the tree it is not balanced it is just like this.

Now, the magic is if you do a random permutation on this and then after that if you from the BST that is what is called randomly build BST and that will be expected balanced tree. So, the height expected height of that tree is log n. So, that we are going to establish. So, for that we need to bring the this randomized version of this BST sort and the comparison between the quick sort they are same, but all these things we are doing to establish that if we have if we have the randomized randomly build BST that is expected that will be the balanced tree I mean on an average expected. So, this is the average case analysis. So, that we are going to show.

Now so, far what we get we get the average depth of a randomly build BST s log n. So, does it mean the height will be log n if the average depth is log n. So, let us see if we can have a tree wherever is depth is log n, but the height is not log n height what is the height of a tree height of a the tree is the maximum depth of the tree that is the height of the tree. So, this does not mean the height is log n. So, what is the example?

(Refer Slide Time: 09:42)

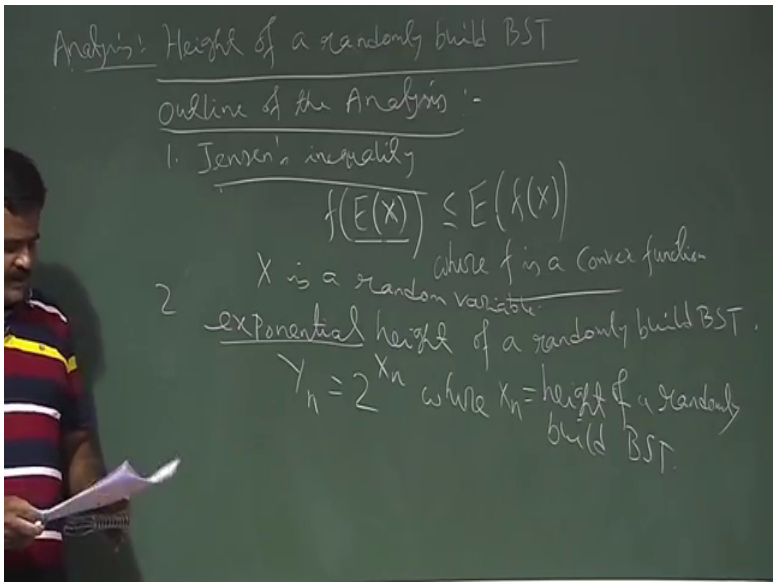


Let us take an example where average depth of the tree is log n, but the height is not log n suppose we have a tree like this. So, almost all the path it is balanced.

Except some branch now suppose this is basically this branch is root over off n and suppose remaining are remaining all the element are balance log n I mean. So, only we have one branch which is more. So, now, all the all are other elements are like this; so, only one branch. So, what is the average depth over here average depth is less than equal to 1 by n into n log n. So, there are n element which are height log n and there are root over of n element which are height root over of n by 2 sort of. So, and this is basically log n.

So, average depth is log n, but the height of this tree is root over n height of the tree is not log n. So, even the average depth is log n height may not be log n by seeing this example, but in this I mean randomly build BST we have to prove that the expected height is log n. So, that need a little mathematics to prove that. So, let us try that. So, let us try to prove that randomly build BST expected height is log n so; that means, we have given some input we just do a random permutation on it we just apply random permutation in then we form the tree BST and then expected that BST will be a balanced tree. So, let us try to prove that.

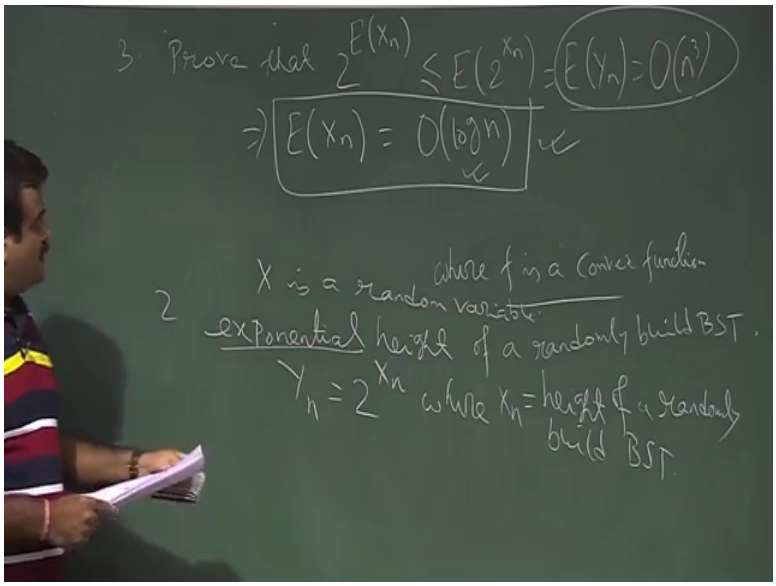
(Refer Slide Time: 11:56)



So, this is the height of a randomly build BST and this is expected value of this is log n. So, this analysis will do. So, this is the outline of this analysis. So, we are going to show this expected value of this analysis. So, what is the outline? So, outline is basically we first prove a Jensen inequality Jensen or we can use this; this is about the convex function what it is telling it is telling expected value of or f of f is a convex we will prove that expected value of this is less than equal to e of f of x. So, this is basically where a fix is a convex function. So, we will prove this convex function and we will defend the convex function this is the first part of the analysis. So, we have to. So, for any function f which is a convex function for that function f of e of x expected value of x is basically e and f will be exchanged less than equal to e of f of x and x is a random variable random variable.

So, we analyse the exponential height of this exponential height of a randomly build BST. So, that is basically we denote by Y n Y n to be 2 to the power x n where x n is the height of a randomly build BST expand x n is the height of a randomly build BST with n elements. So, x n is the height of a randomly build BST with n element when we define we are we are taking the exponential height. So, we define Y n to be 2 the power x n and that will help us to great the expression and then in the next step.

(Refer Slide Time: 15:24)

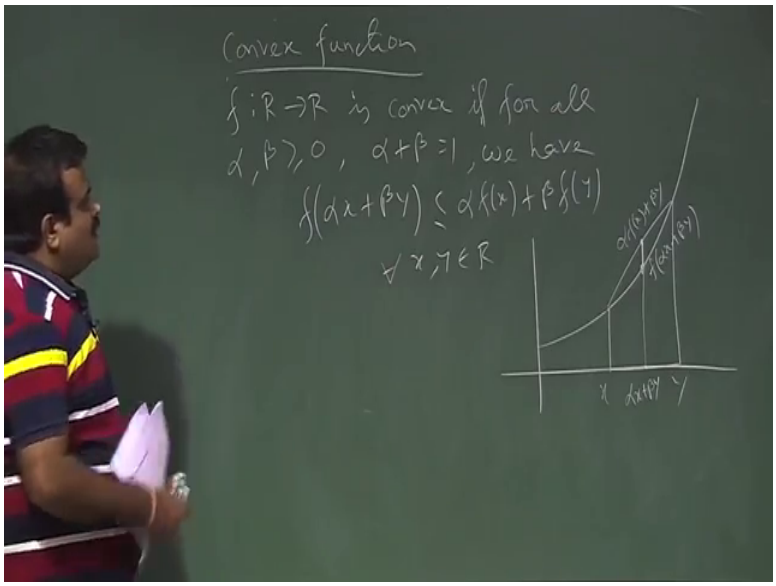


In the final step we prove that we prove that 2 to the power expectation of x n this 2 to the power x is a convex function is basically. So, we will use this Jensen; Jensen inequality is basically expectation of 2 to the power x n which is basically Y n we prove this Y n to be order of n cube.

If we can show this then this implies expected value of the height is basically order of log n this we have to establish finally, to establish this we first prove expected value of Y n is order of n cube if we can prove that then by using the Jensen inequality we can say 2 the power expected value of x n is less than equal to expected value of 2 to the power x n because this 2 to the power x is a convex function and then this is basically our Y n and if we can show this Y n is bounded by n cube then this is telling us expected value of x n is log n so; that means, it is a balanced tree. So, this is the outline of the proof outline of the analysis now let us talk about the Jensen inequality.

So, for that let us define; what is the convex function. So, convex function 2 to the power x is a convex function and for convex function we are using that Jensen inequality the theorem.

(Refer Slide Time: 17:15)

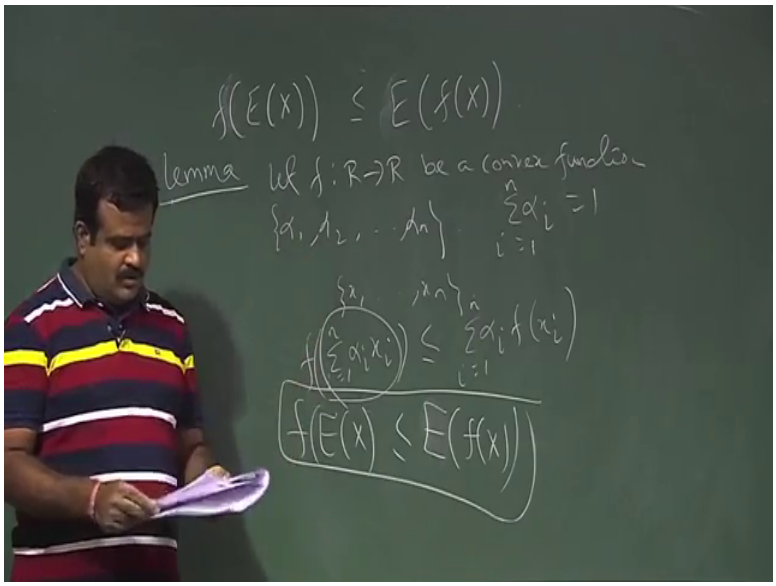


So, if function f from r to r real number set to real number is convex function this is the definition is convex if for all alpha beta greater than 0 such that alpha plus beta is one then we have f of alpha x plus beta Y is less than equal to alpha f x plus beta f Y where for all x Y belongs to r. So, this is the definition of convex function.

A real valid function f is called convex function if for a given alpha beta if for all alpha beta greater than 0 such that their sum is one we have this. So, like if we have a example. So, if we have this function this function is convex. So, if you take 2 point x and y. So, now, this is basically. So, if we take alpha; alpha x plus beta Y where alpha beta is one. So, this is this point is basically here alpha x plus beta y. So, this point if we take the value of this function and this is the; if we take this straight line.

And this is basically alpha f x plus beta f y. So, this is basically should be less than and this is basically f of alpha x plus beta y. So, the curve is like this. So, this must be less than this value. So, curve will be like this. So, this type of function is called convex function. So, if you take any 2 point and the curve. So, if you take any 2 point x Y and if you take any other point then it is the all the point inside must be inside of that curve. So, this type of function is convex function and the Jensen inequality is telling and e to the power can be shown to be convex function and the lemma which is the Jensen inequality this is telling us expected if we have a convex function f then the expected value of. So, f of expected value of x is less than equal to e of f of x.

(Refer Slide Time: 20:02)

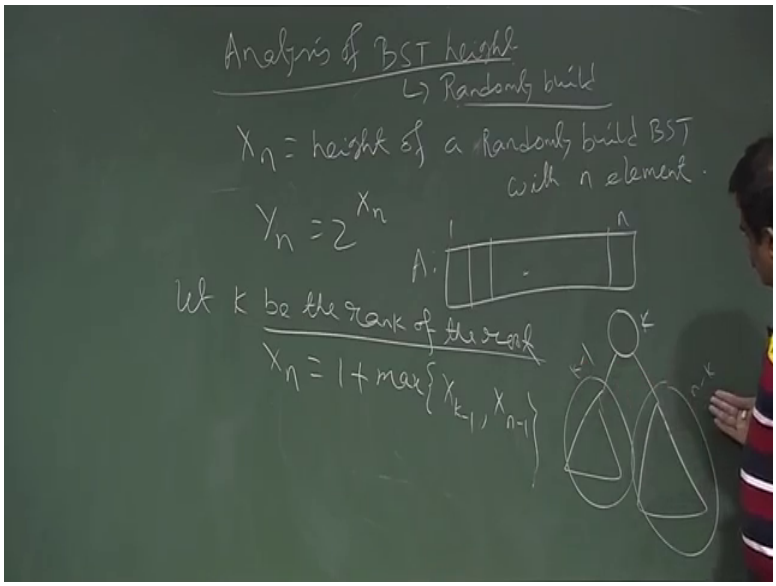


So, how to prove that; so, to prove that we have to use this lemma what is this, so, let f be a convex function f b a convex function and then we have a set alpha one. So, this is generalized function alpha one alpha n there we have alpha beta, but if you have n elements such that summation of alpha is equal to n then if we have the point x 1 x 2 x n this is generalized version of the; which we have seen for 2 points.

this is the n points the f of this prove can be extended by using the earlier 1 f of summation of alpha xi this less than equal to and this can extend for infinity also provided this sum is absolute convergent summation of summation of alpha i f of x i. So, this i is from one to n and this i is from 1 to n. So, this can be proved by. So, this is basically giving us this now if x is a discrete random variable if x is a discrete random variable with this points then this is nothing, but and these are the points and this is the probability then this is basically expectation. So, this is basically f of expectation of x is less than equal to summation of this is the probability and this is. So, this is basically expectation of f of x.

This is the definition. So, this is basically our Jensen inequality and this is true for discrete random variable. So, this is the proof for Jensen and we are not this; this can proved using the induction. So, we are not going to the details of this proof now let us go to the height analysis.

(Refer Slide Time: 22:34)



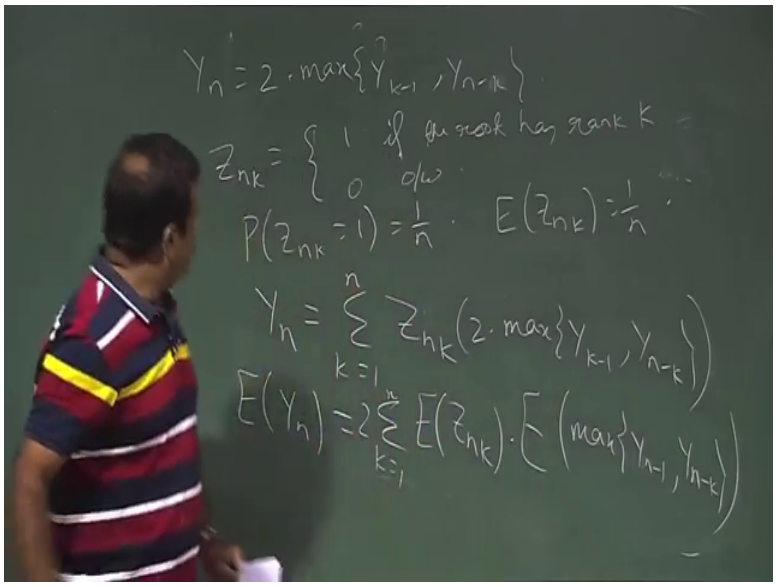
And there we will use this Jensen inequality for 2 to the power x case. So, analysis of BST height randomly builds BST height. So, we have taken x n to be height of the of a randomly build BST with n element.

With n element; so, we have given an array of n elements. So, what we are doing we just permuting these array randomly and then we are forming the BST and x n is the height of that BST after permuting and forming the BST of that BST and we are taking Y n to be 2 to the power x n and our goal is to if you recall the outline of the our goal is to. So, that expected value of Y n is equal to order of n cube. So, now, now what is the proof now suppose? So, x n suppose this is the we are forming the BST now we do not know the we are doing the random permutation we do not know the we are doing the random permutation we do not know the which will be the root we have given n number.

We have given n numbers and we are just permuting it. So, we do not know the; which one will be the root. So, we do not know the rank of the root suppose rank if the rank of the root is k and that is also random variable let k be the rank of the root rank means the position of that element in the sorted array of the root then x n can be written as what x n is basically one plus maximum of this x k minus one comma x n minus k because root is sitting here. So, height is basically. So, we do not know. So, if rank of this is k then we know how many elements are there how many there are k minus one elements and there are n minus k elements over here because rank is k.

So; that means, so, the height of this tree will be we do not know which is maximum which is minimum height of this tree will be one plus maximum of height of this is k x k minus one and height of this x n minus k. So, this is ok.

(Refer Slide Time: 25:33)

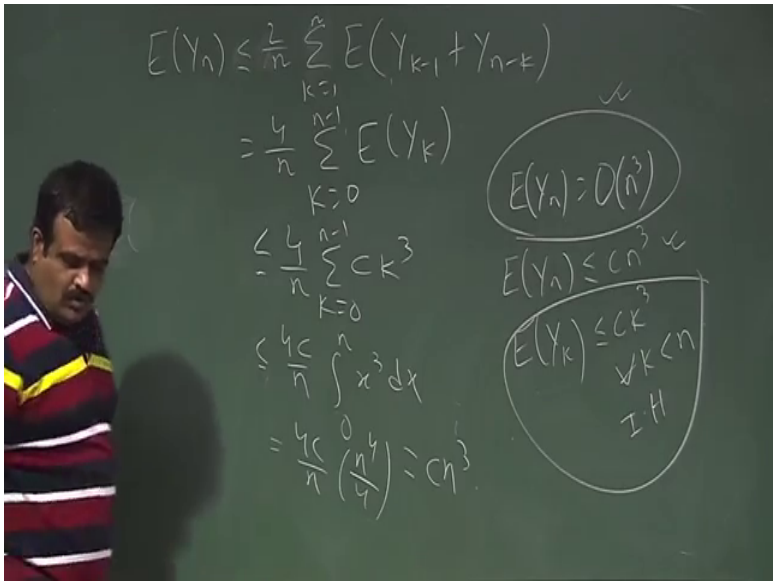


So, now we will take the y. So, Y n will be what. So, from here we can say yn will be 2 to the power of x n. So, 2 of max of x n; so, we will again it will be 2 to the power of that. So, max of Y k minus one comma Y n minus k. So, this is of the form now we will take the help of indicator random variable.

So, we define Z n k is one if the rank of root is if the root has rank k otherwise it is 0 then the probability. So, these are equally likely probability of this is 1 by n. So, the expected value of Z n k is basically 1 by n and. So, Y n will be basically any one of this because rank is k. So, k can take value one to n we do not know which k will occur. So, yn is basically summation of Z n k into this expression 2 of max of Y k minus one comma Y n minus k. So, this type of analysis we did. So, this is our Y n. So, this is from one to n now we take the expectation both sides. So, expectation is a linear function expected value of this and this now again if we take the independentness of these two. So, this will give us expectation of Z n k into expect 2 will come out here. So, k is 1 to n expectation of max of Y n k comma Y this 1. So, let us just give this. So, these we want to further simplify.

So, now this is basically max of this. So, max of 2 numbers is less than equal to sum of their number.

(Refer Slide Time: 28:09)



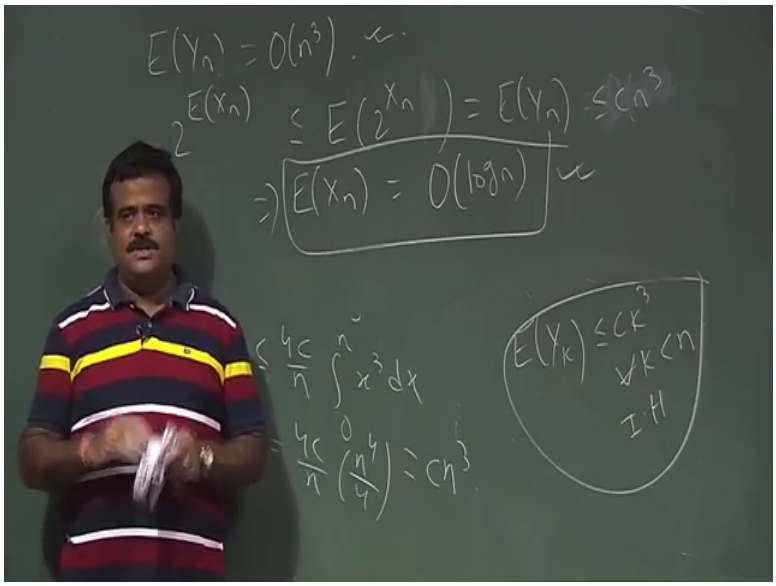
So, this is basically. So, expectation of Y or n can be written as 2 off. So, expectation of this is 1 by n. So, 2 by n and this is basically expectation of summation; summation expectation of Y k Y n minus k because max of a b is less than equal to. So, this must be less than and this k is what one to n and this is basically 4 by n summation of expectation of Y k and this k is now adding from 0 to n minus one because this term is coming twice. So, this is the expression we got. So, how we can further simplify this?

So, this is basically we will prove this with the help of substitution method. So, what we are expecting we are showing that we want to show the expected value of Y n is order of n cube. So, this is our goal to achieve so; that means, expected value of Y n is less than equal to c n cube this is this want to show by the use of substitution method. So, for this we are taking the induction hypothesis c k cube for all k less than n this is the induction hypothesis this table now we put it here. So, this will give us less than equal to 4 by n summation of c k cube k is equal to 0 to n minus 1 by the induction hypothesis step.

Now, this we can again using the integration 4 c by n. So, this will be less than of this integration 0 to n improper integral x cube d x now this if you simplify this will give us 4 c by n into n 4 by 4. So, this is basically giving us c n cube c n cube so; that means, expected value of Y n is less than c n cube. So, by the substitution method we can say

that by the method of induction or substitution method we can say this is true now once this is true then the expected value of. So, so how do we establish we establish that expected value of Y n is order of n cube, this we have proved using the substitution method.

(Refer Slide Time: 31:02)



So, now we take the 2 to the power expectation of y n which is basically less than equal to expectation of 2 to the power sorry x n and this is the by Jensen inequality. So, this is basically expectation of y n and this is basically order of n cube. So, from here we can say expected value of x n is order of. So, order of n cube. So, expected value of Y x n is basically order of less than equal order of log n. So, less than we can use the less than less than equal to c n cube then we take the log both side then it will give us c log n. So, it will be order of log n. So, that is the expected height of the randomly built binary search tree is height is log n. So, it is balanced. So, expected it is balanced.

But the worst case is always like this. So, worst case is always like this because it may happen that we are doing the random permutation the random permutation will give us the sorted array or even sorted array. So, once we from the tree it will be the like this or this. So, the worst case is always quality height, but average case is expected; expected height is log n. So, that is the average case analysis. So, in the next class we will discuss how we can guarantee that we have a balanced tree no; no randomization. So, guaranteed balanced tree. So, they are called balanced tree the like red black tree b tree a real tree.

So, we will talk about red black tree in the next class.

Thank you.