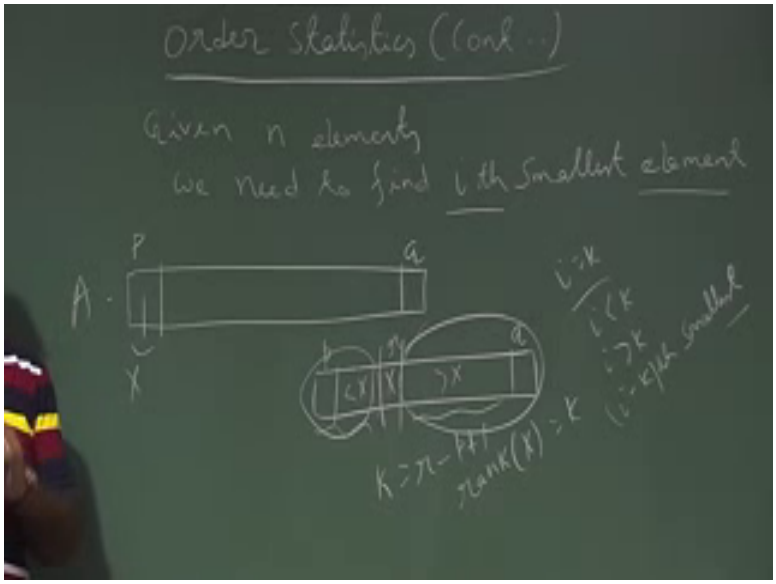
An Introduction to Algorithms
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 19
Order Statistices (contd.)

So, we are talking order statistics problem the problem is to find given n numbers.

(Refer Slide Time: 00:27)



And we need to find out i th smallest element. So, this is the problem for order statistics. So, i is any index form 1 to n. So, if i is equal o 1 this is minimum if i is equal to n this is maximum in if i is equal to n by 2 order then it is with the; it is called median. So, we have seen the select algorithm basically we are using the partition algorithm.

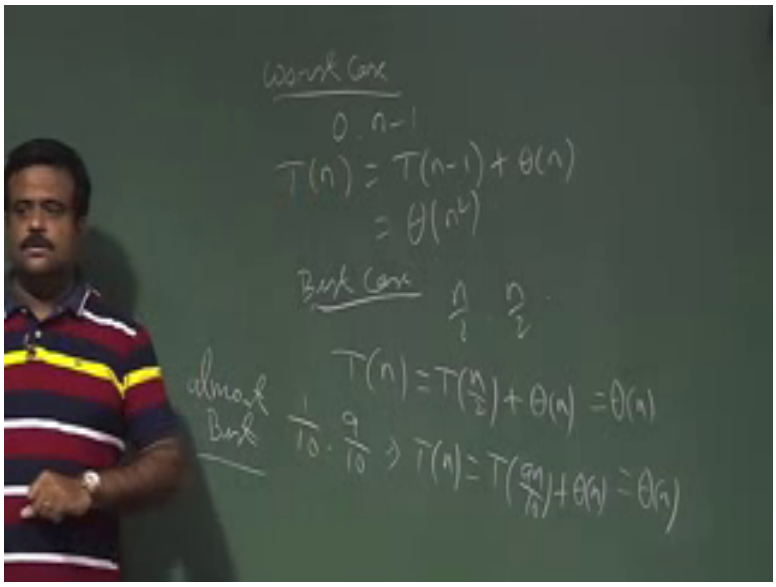
And we partition the; we have given this array of numbers. So, say this is A array. So, we choose this as a pivot element x. So, this is say p to q and then we call the partition then x will be sitting somewhere here and this is the index of x where it is sitting and this is p this is q and then. So, all the elements over here are less than x all the elements over here are greater than x. So, now, this is k if we choose k is equal to r minus p plus 1 then k is basically rank of x.

Now, if i is equal to k then we got the our it h smallest element otherwise if i is less than k this is the divide and conquer technique if i is less than k then we know our i th

smallest element is here then again we call the same function this is the conquer step on this sub array otherwise if i is greater than k. Then we know the it h smallest element will be here, but we already release the k th smallest element. So, we need to look at this sub array with i minus k th smallest element.

So, this is the new i for if you have to look at this. So, this was our select algorithm now this; the time complexity of this will depend on the partition. So, how the partition will be over this array I mean if the partition is good partition I mean rather if the pivot element is minimum or maximum then it is a bad partition then it is. So, worst case we have seen worst case we have seen.

(Refer Slide Time: 03:01)



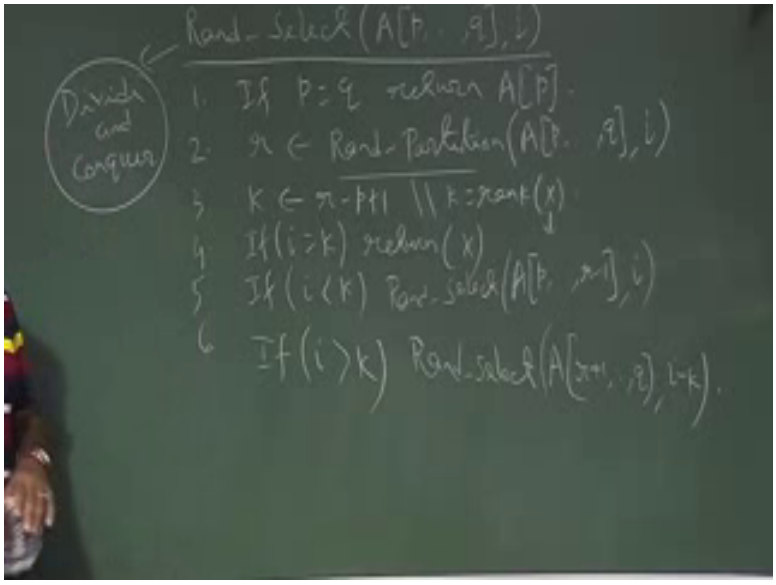
If the pivot we choose always minimum or maximum then the worst case will be 0 is to n minus 1 and in that case the recurrence is theta n. So, this is our recurrence and this is the arithmetic series. So, this will give us order of n square and what is the best case best case is basically if the partition is n by 2 is to n by 2 then we have seen the recurrence is T n is basically.

So, T n by 2 plus theta of n is the cost for the partition algorithm and this by master method of case 3 this is order of n and even if we have almost best case write if we have partition like 1 by 10 is to 9 by 10 then also we have seen the recurrence is T n is equal to T 9 n by 10 plus theta of n. So, this is again will give us theta of n. So, this is the almost best case.

So, the lucky case so, but the worst case is this? So, now, we will talk about a randomized version of this algorithm this is our select algorithm this is divide and conquer approach we choose we divide the array into 2 sub array and we put the pivot in some by its position, its position and then we call the we conquer by calling the again the same function select either on the left side or right side depending on the value of i.

But this is the analysis of that code depending on the partition it will give us the worst case or best case lucky case unlucky case. So, let us talk about randomized version of this. So, if we know that.

(Refer Slide Time: 05:19)



So, in the randomized version of this, so, this is called randomized rand select. So, we have given a array and we have to find out the it h smallest element.

So, here the idea is to choose the pivot element randomly. So, that there is a chance that the partition will be the good partition in some of the sub sequence cases. So, in that case we can say we will. In fact, we will prove that the expected run time will be order of n. So, what is that code; code is if p is equal to q then we just return A p else; what we do we just call a partition; now here we call the randomized partition A p to q.

So, what is this randomized partition? So, in our original partition algorithm we if you the quick sort our quick sort algorithm in the partition algorithm. So, this is the array A p to q. So, we choose A p as a pivot element this was our original partition algorithm, but

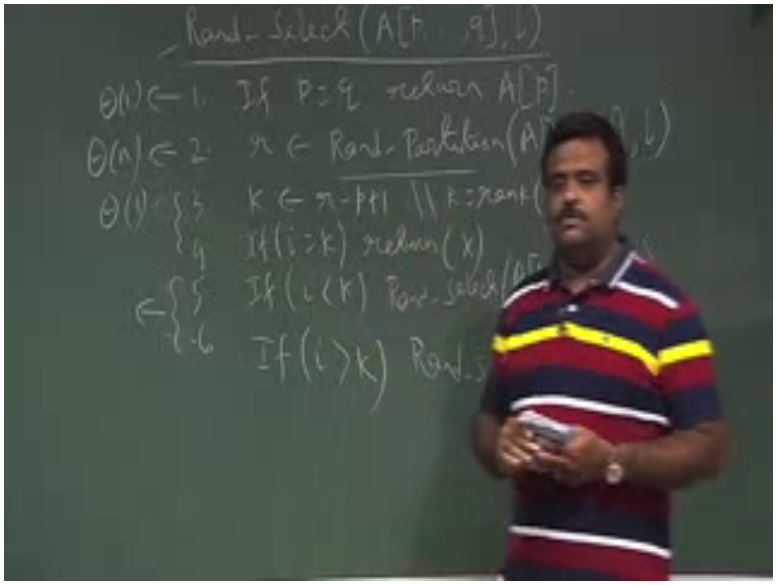
in the randomized version of the partition algorithm any one of this index will be the pivot element. So, like pivot. So, any one of this index we can choose as a pivot element if there are n elements. So, they are equally likely to come.

So, we choose a index randomly from this p to q and then we choose that element as a pivot element. So, that is the randomized version of the quick sort and it will return the r r is the r is basically the index where x is putting after calling the pivot and then we choose k to be r minus p plus 1 k is basically rank of x x is the pivot element which is chosen randomly now the code is similar if similar than the select code we discussed in the last lecture. So, if i is equal to k then we return x that is our it h smallest element else.

If i is less than k then again we have to call this randomized select. So, is less than k means we have to look the it h smallest element in this sub array. So, we call again randomized select on this sub array left sub array A p to r minus 1 with the index i else if i is greater than k then we have to call the rand select on the right sub array, but we already explore the k th smallest element. So, now, we have to get the i minus k th smallest element a r plus 1 to q with i minus k.

So, this is the divide and conquer approach divide and conquer. So, basically we divide the problem into 2 sub here it is one sub problem we either look at the left sub tree or left array or right array.

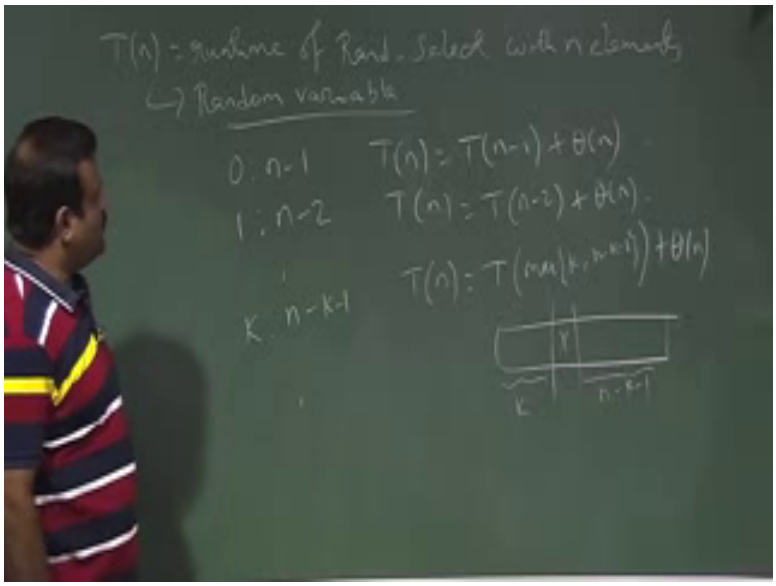
(Refer Slide Time: 09:26)



Now what is the time complexity? So, time complexity will depend on the partition. So, worst case we have seen. So, this will take the order of one time this partition will take linear time and. So, this 2 will take now this, this and this, so, these 2, so, either one call.

So, it will depend on the nature of the partition if we are lucky if the partition is good partition if the pivot is good pivot then we know this is either n by 2 or it is some portion of the some fraction of the n, but if the worst case it is n minus 1. So, depending on the partition it is basically give us the time, but worst case time complexity is always n square, but this is the randomized version. So, we want to look at the average case analysis. So, expected run time. So, that we are going to do now.

(Refer Slide Time: 10:32)



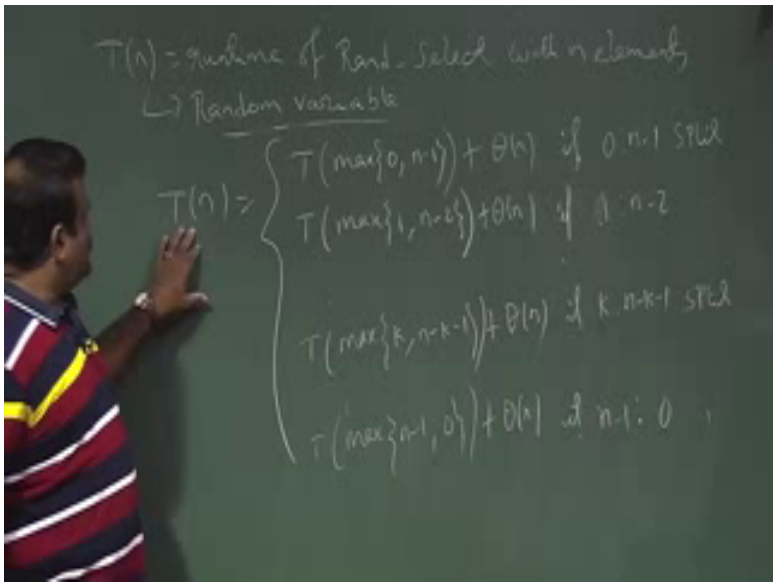
So, for that let us denote the time complexity as T n T n is the run time of rand select with n element now this is randomly chosen pivot. So, the partition we do not know which partition will come. So, that is why this is basically a random variable this is basically a run time variable. So, if the partition is say 0 is to n minus 1 then we know T n is basically T 0 plus I mean T n minus 1 plus theta of n.

So, if the partition is 1 is to n minus 2 then we know T n is basically T of n minus 2 plus theta of n and if the partition is k is to n minus k plus 1 n minus k minus 1 then the dot dot dot then the T n will be basically T of maximum of this 2. So, it is basically partition is this means we have x is sitting here and there are k element over here and k minus n

minus k minus 1 element over here. So, depending on the value of k we are talking about worst case run time. So, you go for the maximum size

So, maximum is depending on the value of k it will be a max; max of k comma n minus k minus 1 plus theta of n is the time for partition sub routine.

(Refer Slide Time: 12:45)



So, this way we continue. So, we can write T n in the functional form like this. So, depending on the partition, it will be of the particular that particular form. So, T n is basically in the functional form it is basically T of max of 0 comma n minus 1 this is for symmetric we are writing in the max is n minus 1.

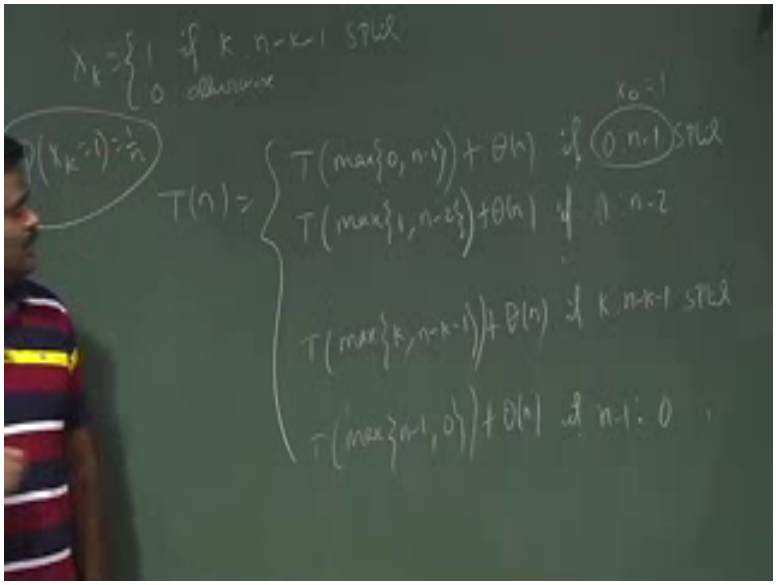
But still for the symmetry we are writing this if the split is if the split or partition is 0 is to n minus 1 split or it is T of max of 1 is to n minus 2 if the split is 1 is to n minus 2 like this dot dot dot T of max of k comma n minus k minus 1 plus theta of n for the partition. So, if the split is k is to n minus k minus 1 dot dot dot T of max of n minus 1 comma 0 plus theta of n if the split is n minus 1 comma 0 split.

So, this is the T n form of the T n in the functional in the in this form now T that is why T n is a random variable because we do not know which partition will occur because our pivot is randomly chosen. So, before running this we do not know which one is going to be a pivot. So, that is may; that means, we do not know which partition will occur, but at least one of this partition has to be occur, but we do not know which one. So, that is why

T n is in this functional form now we want to take the expectation of this T n we want to calculate the expected value of this T n and that we are going to show it is linear.

So, for that we want to take this functional form to algebraic form. So, for that what we need we need to take help of the indicator random variable. So, now, we define indicator random variable. So, x k, so, let us just.

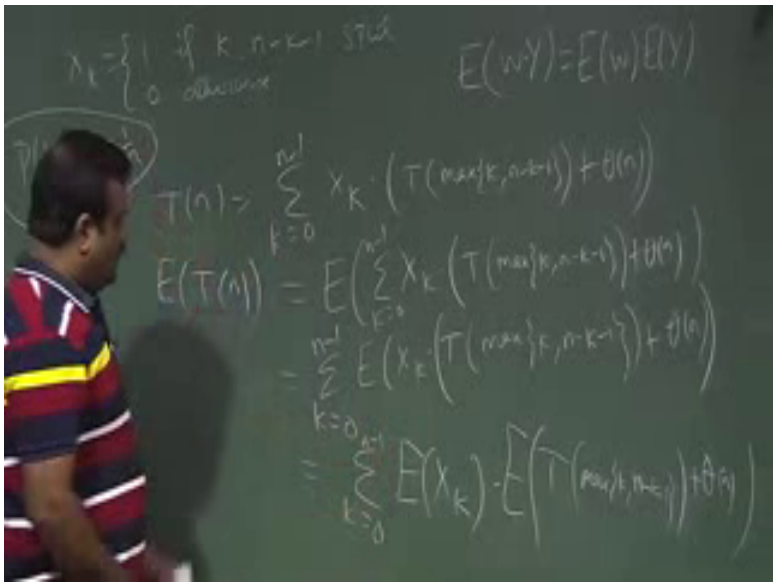
(Refer Slide Time: 15:17)



Let us just define the indicator random variable x k x k is 1 if the split is k is to n minus k minus 1 split 0 otherwise. So, this is the. So, one of the escape value is 0. So, if the partition is this then this is x 0 x 0 is one then remaining all other x value is 0. So, now, one of the partitions will occur and they are equally likely to occur.

So; that means, probability of x k equal to 1 is 1 by n because there are n partitions. So, that is why it is 1 by n. So, now, we want to write this in terms of algebraic form because now we have a indicator random variable. So, 1 of this x k 1 of this partition will occur; that means 1 of that x k is 1 remaining as 0.

(Refer Slide Time: 16:26)



So, basically we can write T n as in the some form summation of x k that particular one of the split is one into that corresponding t.

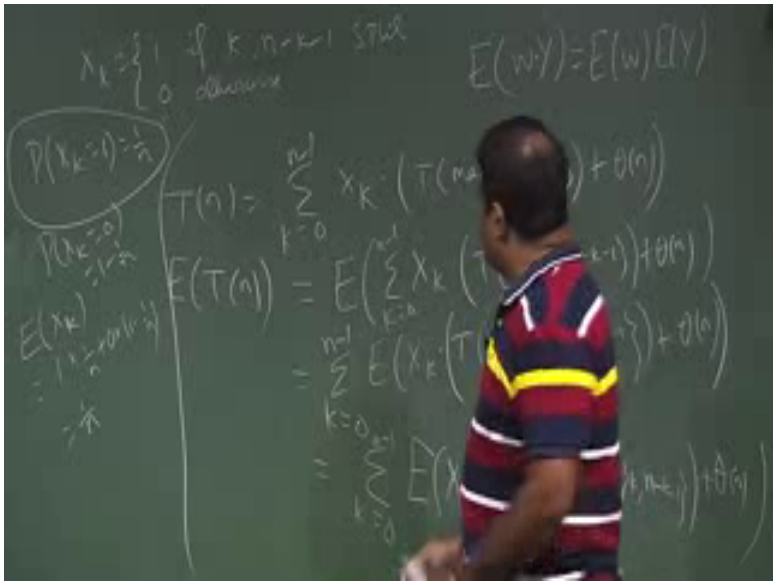
So, this is basically T of max of T comma n minus k minus 1 plus theta of n. So, this is basically our x algebraic expression and this T k is running from 0 to n minus 1. So, 1 of the partition will occur and that corresponding x k is 1 remaining x ks are 0. So, that is that is why we can write this in terms of this. So, now, we want to take the expectation on both side. So, if you take the expectation of T n.

So, this is basically expectation of this now summation of x k into T of max of k comma n minus k minus 1 plus theta of n. So, this k is from 0 to n minus 1 now expectation is a linear function we can take the expectation inside, so, summation of expectation of x k into T of max of T comma n minus k minus 1. So, this is the closing bracket of this plus theta of n. So, this will come with x k into this n this and this k is from 0 to n minus 1.

So, now we want to take the; this is x k into some random variable so; that means, we have something like that w y we have 2 random variable w and y expectation of w into y. So, if they are independent then we can written as expectation of w into expectation of y. So, if y and w are independent random variable independent why independent ness is coming here because here we are choosing the pivot element randomly. So, for that we need to generate the random number in the in the subsequent steps also.

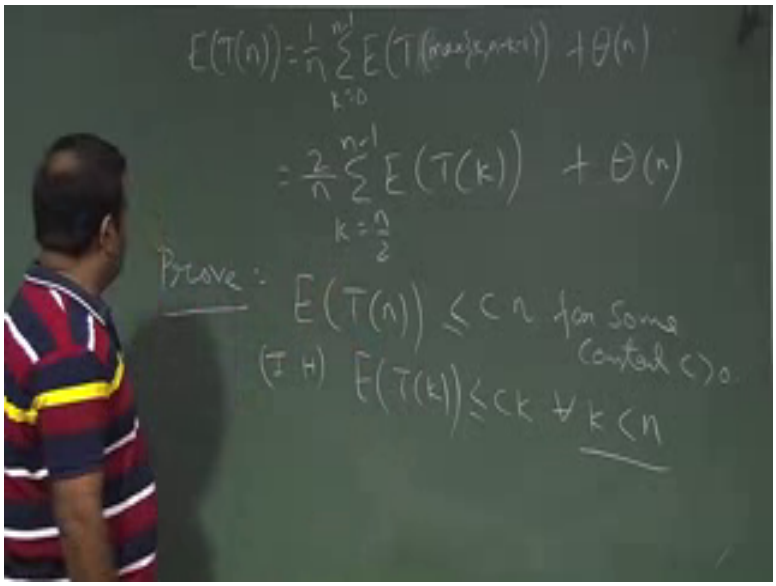
So, this all choice of random numbers generation are independent. So, that sense it is coming the independent ness is coming. So, if we assume that independent ness then this can be written as summation of expectation of x k into expectation of T of this thing maximum of k comma n minus k minus 1 T of this plus theta of n and this k is varying from 0 to n minus 1 now what is the expectation f x k.

(Refer Slide Time: 19:47)



Now, probability of x k is equal 1 this and probability of x k is equal to 0 is basically 1 minus this. So, this is a 2 point; this is a binary random variable. So, expectation is basically 1 2 value 1 and 0 1 into its probability plus 0 into its probability. So, this is basically 1 by n. So, expectation of x k is basically 1 by n. So, we put this value over here 1 by n. So, this will basically give us erase this.

(Refer Slide Time: 20:28)



So, this will give us expectation of T of n. So, this is 1 by n we take outside summation of this thing. So, expectation of this plus this and this is nothing to do with the expectation. So, this will basically give us again theta of n. So, this is basically summation of expected value of n T of T of max of k comma n minus k minus 1 plus theta of n this is from k from 0 to n minus 1. So, now, how we can simplify this further. So, this is basically.

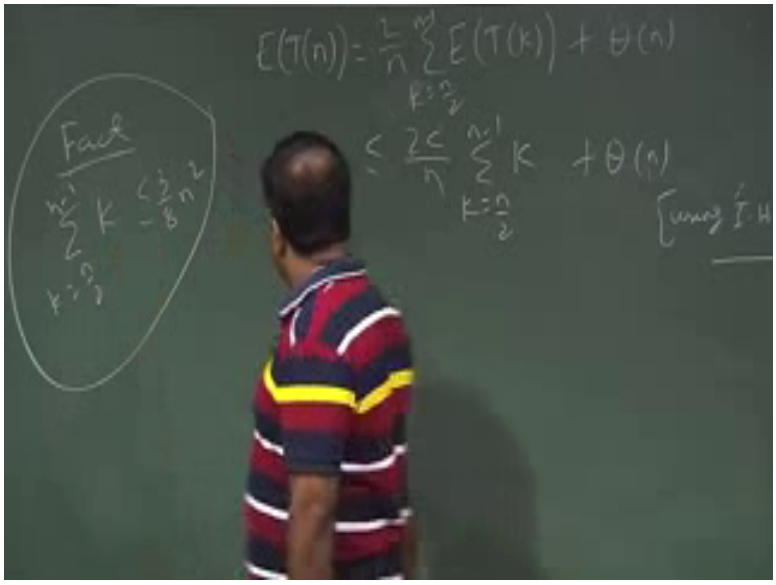
So, let us just say try to simplify this further now this is basically we are taking the expectation of T of maximum of this, now k is varying from 0 to 0 to n minus 1 now if k is up to n by 2 then this maximum will be n by 2 i mean. So, these can be written as. So, these can be written as this is 2 by n summation of expectation of T of k where k is varying from n by 2 to n minus 1 plus theta of n because see when k is equal to 0 then the maximum is n minus 1 when k is equal to 1 then maximum between 1 and n minus 2 is n minus 2.

So, basically, so, if k k is varying from 1 2 like this n by 2 then n by 2 plus 1 to n now if k is 0 then maximum is n minus 1 k is 1 maximum is this like this. So, k is if k is n n by 2 then the maximum will be n by 2 again from here also we have maximum n by 2 to n. So, basically we have twice term this sum is basically we have 2 of this is this clear. So, now, we have to simplify this mode. So, this is basically the expression we got for the expected run time now how this is linear we have to show that to show that.

So, w have to prove that this is we want to prove expected value of T n is basically big o of n. So, for that we want to take this as c of n some constant for some constant c greater than 0. So, this how we can prove this. So, now, will prove this by method of induction the substitution method now we assuming that this is true this result is true this is the induction hypothesis. So, we assume this result is true for up to n.

So; that means, expected value of T of k is basically less than equal to c k for all k less than n this is our assumption and then we prove that this is true for n also. So, this is the way this is the substitution method. So, we have this expression.

(Refer Slide Time: 24:36)

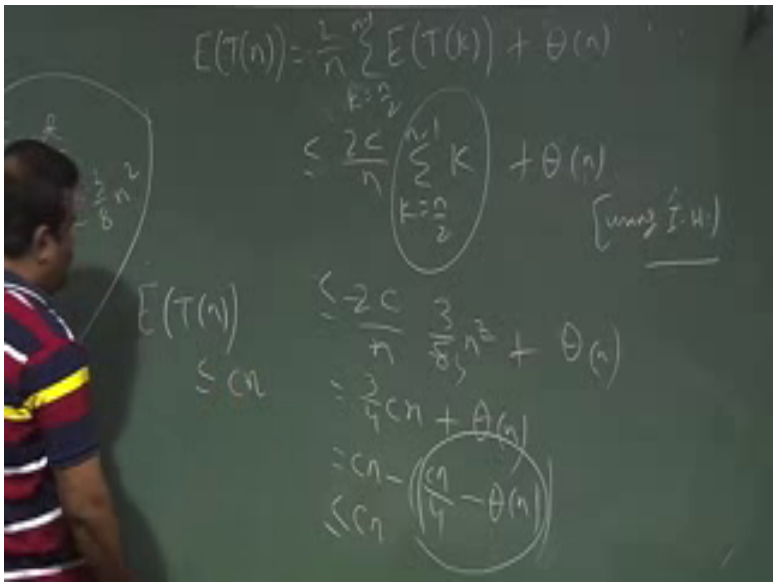


So, this is basically the expression we have 2 by n summation of k is equal to n by 2 to n minus 1 expected value of T of k plus theta of n outside. So, now, we will use this induction hypothesis. So, we will use this induction hypothesis.

Because this is the all the k value n by 2 to n minus 1 these are all less than n. So, this will give us 2 c by n summation of summation of k k is equal to n by 2 to n minus 1 plus theta of n this is by using induction hypothesis. So, this is by using the induction hypothesis now we have to prove that this is less than c of n. So, to show that we need to take a inequality which is which is basically telling us this is a fact.

So, summation of k k is equal to n by 2 to n minus 1 is basically can be shown as 3 by 8 n square. So, this needs the; prove, but you can prove it by again by induction. So, this expression this inequality we are going to use if we. So, this sum is similar to this sum.

(Refer Slide Time: 26:11)

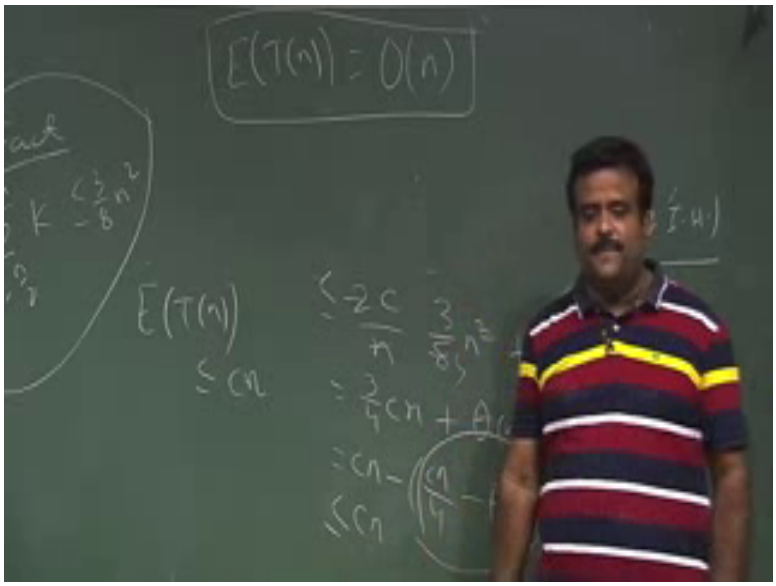


So, this is again 2 c by n into 3 by 8 n square plus theta of n. So, this is basically giving us what. So, this is 4 this n n cancel 1 n.

So, 3 by 4 c n plus theta of n, so, now, we have a control on c; c is the constant we can play with. So, we choose c in such a way that this should be less than. So, we choose c such a way that this should be less than. So, this we want to write as this c n minus c n by 4 minus theta of n. So, these expression if this is positive then we can write this is less than equal to c n, but to have this positive we can choose c is large such that it will be positive because this is theta of n means it is some expression in n some function in n.

So, we can choose c in such a way this will be the positive. So, this is basically less than n. So, this is the proof.

(Refer Slide Time: 27:36)



So, expected value of T of n is basically less than c of n. So, this is basically telling us by the method of induction the expected value of basically big o of n. So, this is the average case analysis of randomized version of the say randomized select, but this is the average case. But the worst case is always n square because even we are choosing the pivot randomly it may happen that always that pivot is coming as minimum or maximum all though this is random choice, but that chance is there.

So, the worst case is always order big o worst case is always 0 is to n minus 1 partition, but this is the average case analysis for this randomized version of the select.

Thank you.