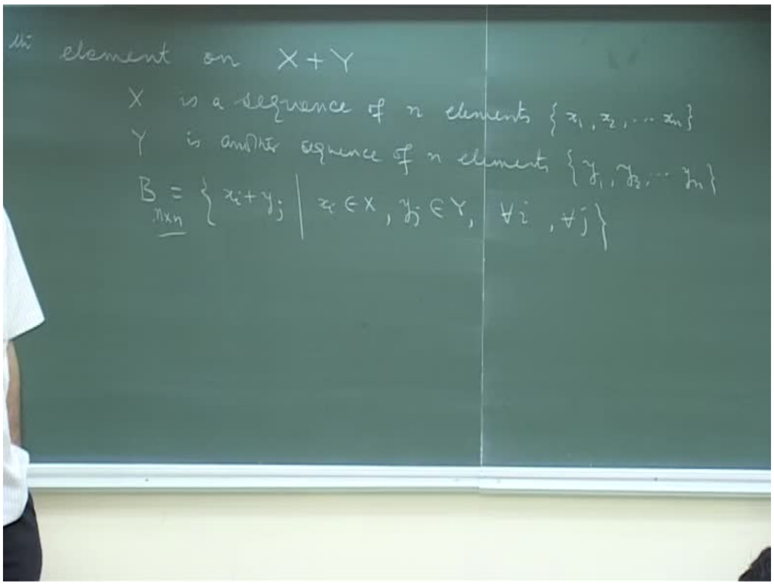
Parallel Algorithms
Prof. Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 16

Now the problem we discussed what is that case elements on sequence or set of n elements that we discussed already on EREW model and CREW model, agreed?

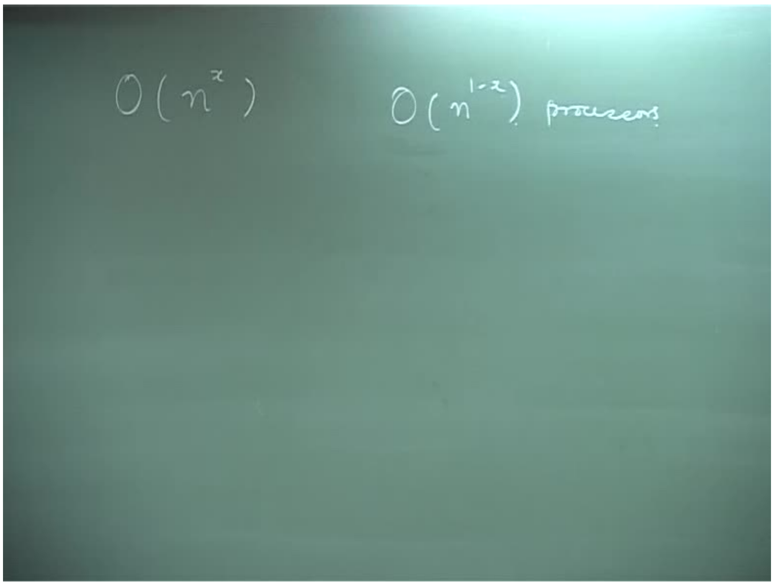
(Refer Slide Time: 00:33)



Yes, now today we will be discussing finding the k eth element on x plus y, where x is a sequence of n elements say x 1, x 2, x n and y is another sequence of n elements. This is for simplicity we have taken that size is same; otherwise, you can think about one is of sequence size n, another one is having sequence of size n. Now x plus y is y is the Cartesian sum, so x plus y will give you a matrix of size n square, right. So, B is a matrix, which I tell x plus y j, x i belongs to x and belongs to y and for all i and for all j. So, this B is an enclosure matrix, you really do not know? Yes, now I have to find out the k eth element from this B, how to find it?

Now, one way could be that use your whatever algorithm we have discussed with elements for finding the k eth element on n that can be used, right; it can be instead of n you have n square elements. So, time complexity would be partial order; we have used some algorithm, right, n to the power 1 minus e. Last class you check finding the k th element, there is a partitioning element which divides k sequence, l it is not there?

(Refer Slide Time: 03:32)

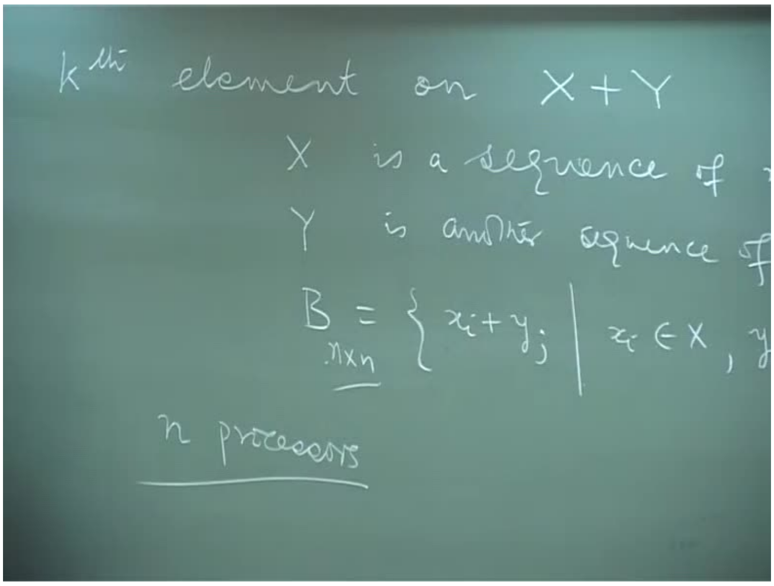


Student: Order of n to the power x.

Order of n to the power x using order n to the power 1 minus x plus that, like that, is it not? So, this algorithm also can be used here to find the k th element, but you know it has some properties which are not being used to find the k th element if I use this one; sum properties means that x side is added to all y, right. So, impact on that row is fixed almost, use a shift by x i value, right. So, that information is not taken into account here, right. So, that information if you can take then possibly we will be getting would be discussing how we will be getting the better results.

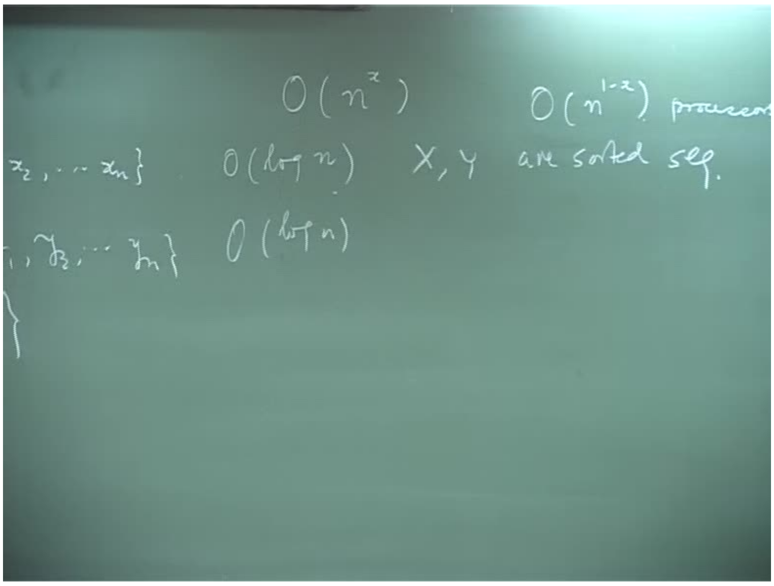
But another thing, what we did in this is that instead of using the median element to partition into n. We took the three elements and second largest element; the middle of this three elements. We have considered it as the partitioning element and size is, if you remember instead of discarding 25 percent, at least 25 percent discarding I can do on usual k eth element. We will be showing then, no, we can discard more than that if I use the second largest element of the three elements and by that process you will be reducing your searching area and it becomes the little faster than the usual thing, but complexity wise it will not change.

(Refer Slide Time: 05:43)



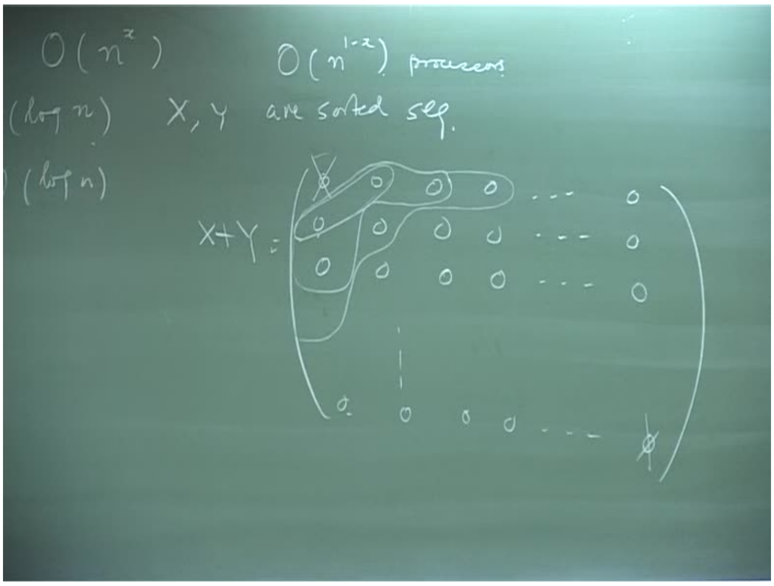
As this algorithm is on shared memory model with n processors, now a little background I want to tell that if what happens; let us assume them x is sorted or if it is not sorted you can use your n processors to sort it. So all x is not sorted; you can use your n processors to sort it, right and which takes order log n time by the most recent algorithm we can use then it takes order log n time.

(Refer Slide Time: 06:39)



Similarly, I sort y and using n processors it takes order log n time that is the cost optimal algorithms you have. So, in terms of generative we assume that they are sorted or if not you sort it and you get. So, x is sorted, y is also sorted, right.

(Refer Slide Time: 07:02)

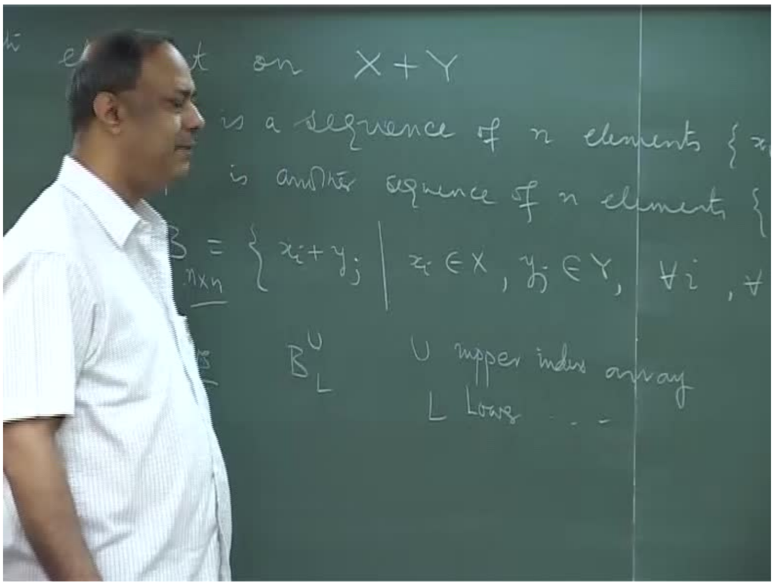


Now what happens if x and y are sorted? This is your enclosure element. Now suppose I am interested for finding the minimum element, in that case it is very simple; this is the minimum element. I have to find out the largest element, this is the largest element, do you get it? Now if I have to find out the second minimum element; second minimum element is this one, one of these two, right. So, in constant time you can find out. This cannot be the second largest element, because this gives you the guarantee all these elements will be smaller than this elements, because x is sorted and y is sorted, right. Now if suppose I have to find out the third largest element, so which are the probable candidates? This can be a third largest element, this can be a third largest element, can it be a third largest element? No, right, but this can be a third largest element; this can be a third largest element. So, this is your searching zone for third largest element, agreed?

Now in order to find out the fourth largest element so this can be like this. So, in order to find out for small value of k, for small value of k you do not have search all these. So, this is true for k smallest elements or k th largest elements, right. If you are looking for k th smallest element you will be searching from this side, if you are looking for k th largest element k is very large n th element maximum second maximum, second third maximum. Then we have to search into from this side, or if you want to do it, no, no I want to do it only for the smallest element instead of making that x positive you negate it, and then you will get then it will become the smallest element, yes or no?

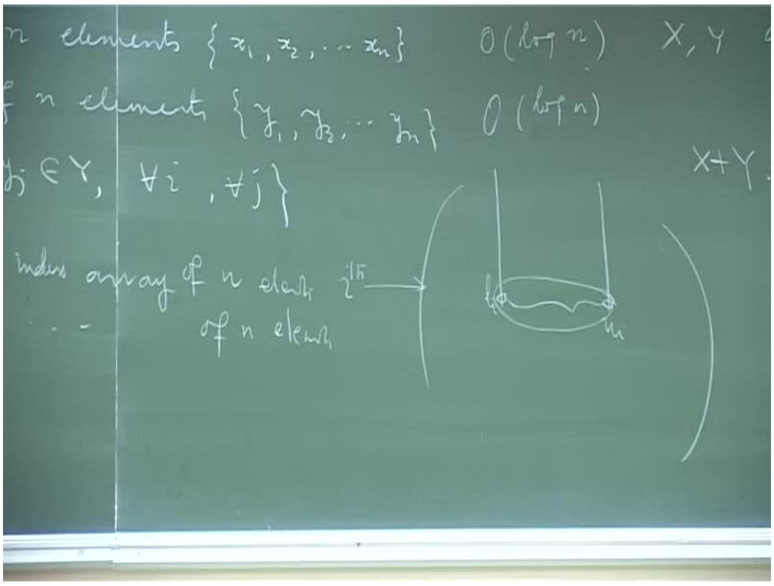
If you negate it then the smallest element become the largest element, largest element become the smallest element. So, that can be done easily. So, without any loss of generality you assume that you are looking for the k th element where k is less than equals to n square by two. So, you are finding the k th smallest element and you have arranged them in increasing order. So, you are looking for this upper diagonal part not the lower one, and also if we have already discussed this part of when k is small you deal with constant amount of time to find out the k th smallest element, yes. Now what happens if k is not small, how can you do it? But you observe that it is very good property. These properties, can you make you use of it? That is the only thing we are looking for.

(Refer Slide Time: 10:43)



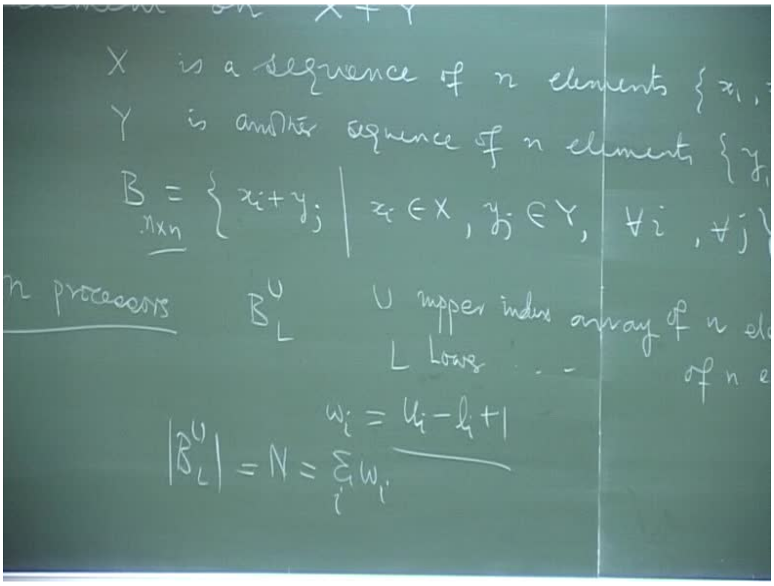
So, let us define the B U L is a subpart of the matrix, where U is telling the upper index array and L is lower index array.

(Refer Slide Time: 11:13)



What it means that U I, this is the i th element i th row, you have L i and you have U i, right, array of n elements. So, in the i th row you have the two index L i and U i which indicates that in the B U L i th array contains this information this many elements information, because this is required, because you have to discard some part of the matrix. So, the part of the matrix you want to consider under consideration should be defined by these two parameters. L i is the starting index of the i th row and U i is the upper index of the i th row and you are assuming these elements are possible candidates for the k th element, okay.

(Refer Slide Time: 12:40)

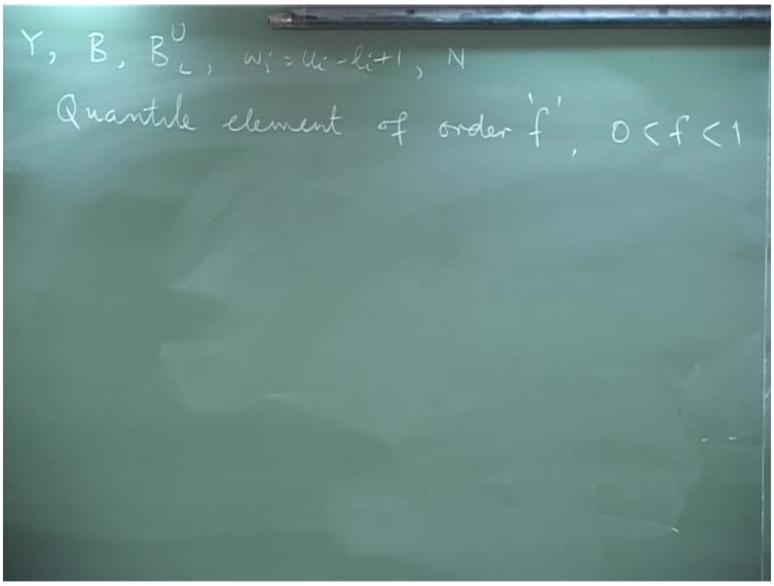


So, in the i th row the number of elements are considered for possible candidate of k th element is U i minus l i plus 1, agreed. And let us assume this is the w i and w i is the weight assigned to the i th row or w i is the number of elements in the i th row which can be considered as the possible candidate for the k th element, agreed. Now N is summation over w i for all i which tells the total number of elements which are probable candidates to become the k th element is there, right, the n is the total number of elements which are the probable candidates for k th element. So, n is nothing but the number of elements in B U L, is that okay.

(Refer Slide Time: 13:59)

So, what we have done? We have defined x, we have defined y, we have defined B, we have defined B U L, we have defined the w i equals to U i minus l i plus 1, and then we have defied N.

(Refer Slide Time: 14:17)



Now you have a word quantile element of order f, right. This is the general term you have used in several places in your statistics or some type of course, right, yes, no? No, quantile element, middle element you know; first quantile you know.

Student: Quantile.

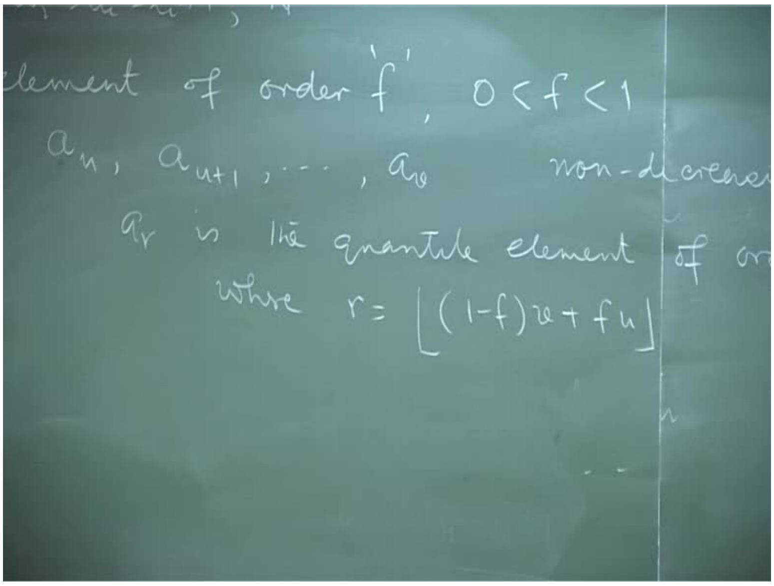
Quantile, you have to know first quantile you have not used; anyway in your daily life also third quantile, first qunatile. First quantile is 25 percent, right; third quantile that in business also you do it.

Student: Yes.

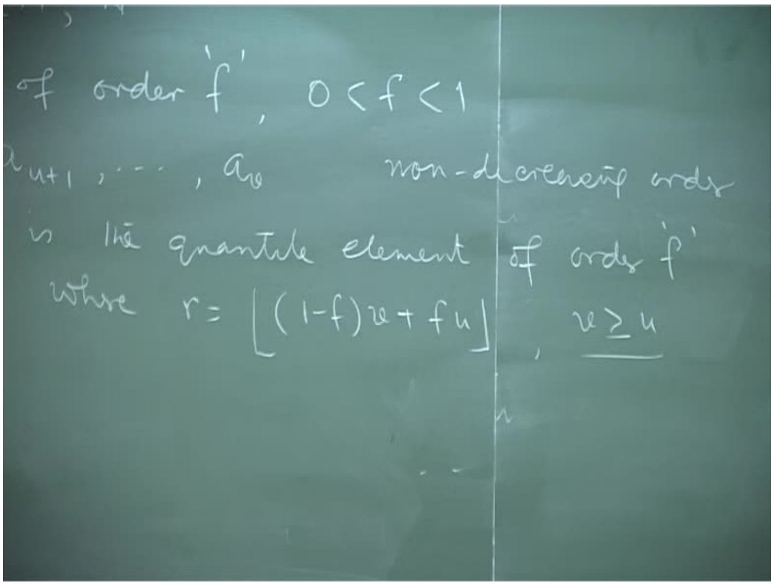
So, before that it is a quantile. As I told you by that what is by target instead of using the median element who wants to introduce the second largest and the second smallest element of three elements as to consider as the partitioning element. But which will divide the matrix B into the three parts less than equal and greater than parts; and we want to see what should be the value of f, so that you can discard maximum number of elements at each iteration, you understood right. In the case of k middle element, you have shown that middle elements gives you the guarantee that 25 percent of the total number of elements you will be able to discard at every iteration, agreed or not, that you will discuss.

Now, what we want to do? We do not want to lose the middle element or link; we want to take the three elements, and we want to use the middle of these three elements as the partitioning elements to reduce the size of the matrix. This gives you the guaranteed 25 percent because middle element is lying there. So, 25 percent you can, but can you do better than that if I use this second of these two three elements. So, you have to learn first; let us define what is quantile elements of order f, and then we will define how to how to go about it, we will discuss through stages.

(Refer Slide Time: 17:02)

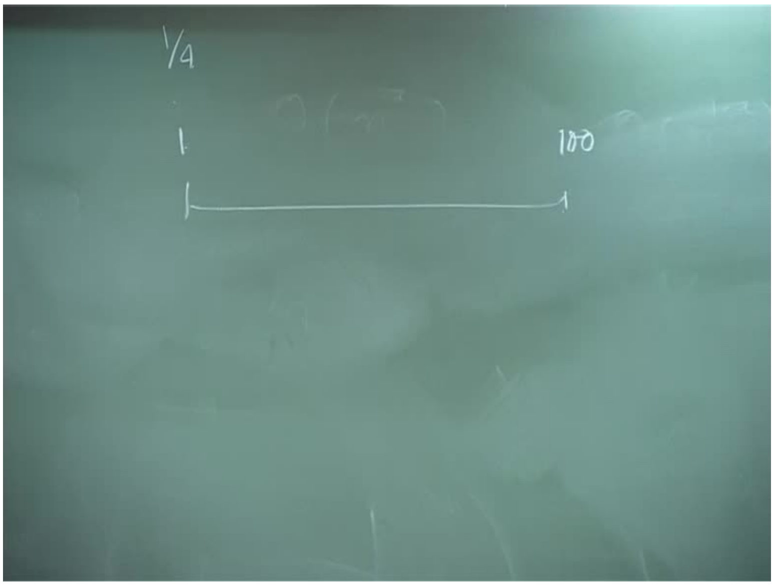


(Refer Slide Time: 17:22)



Let us assume that you have the sequence of elements a u, a u plus 1, a u a b in non-decreasing order. Then a r is the quantile element of order f where is 1 minus f v plus f u where v is greater than u greater than equals to u. Suppose if f is half, what it means? It means that u plus v in your lines plus the mid length.

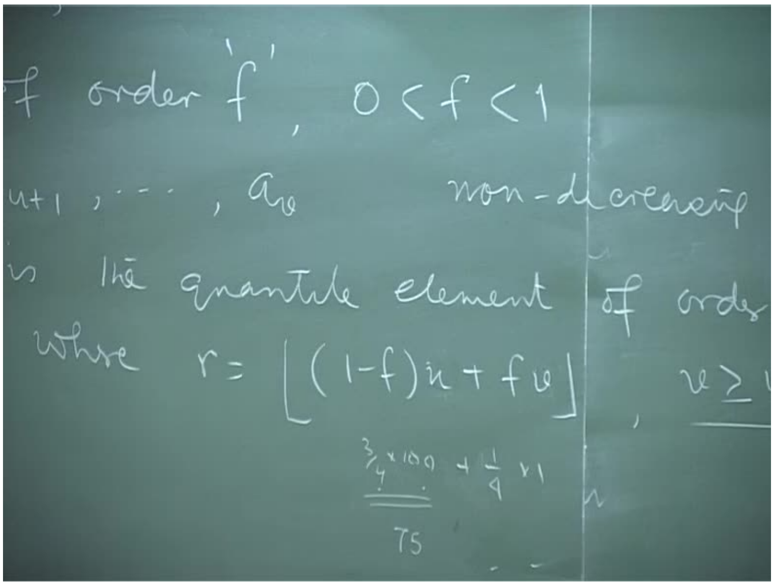
(Refer Slide Time: 18:37)



Suppose if f is 1 by 4 and suppose U is say f is 1 by 4 and say U is 1 and, say, V is 100. So, what it means?

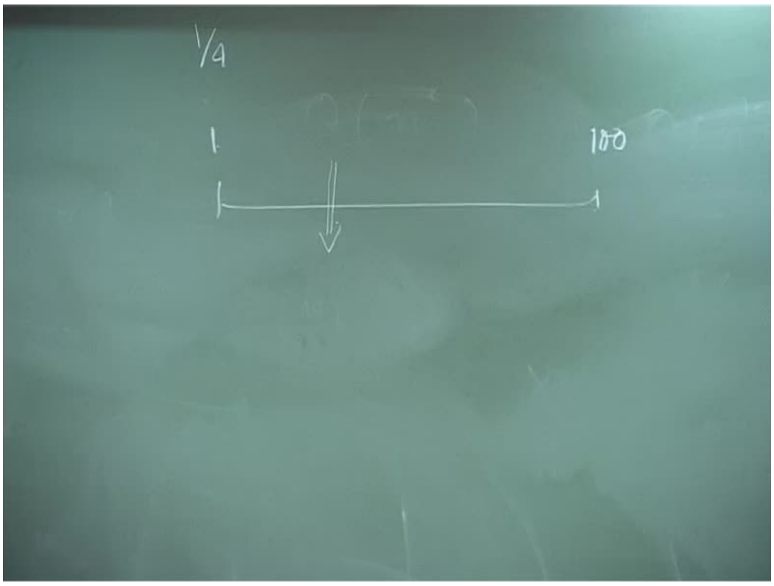
Student: First quadrant predictable.

(Refer Slide Time: 19:11)



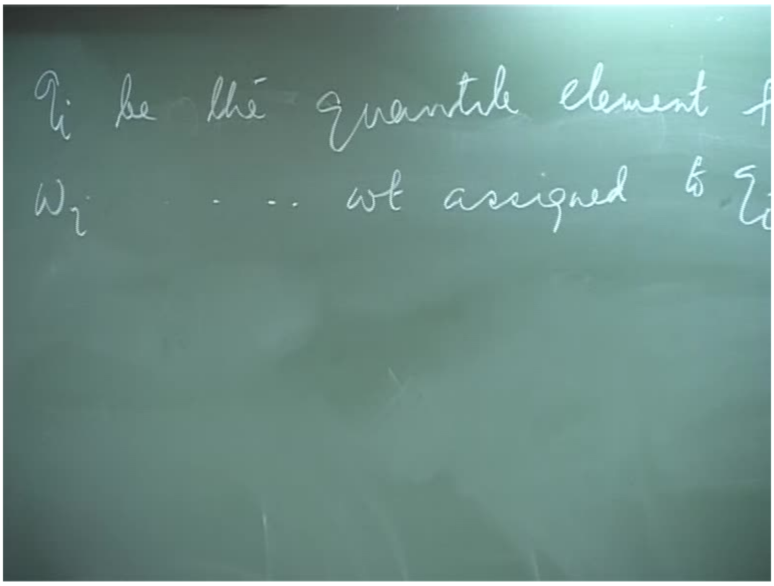
f is 1 by 4, so it is 1 by 4 and this is k by c by 4 into 100 plus 1 by 4 into 1. So, what it tells? These are defined in different ways; otherwise, I think that will give you. In that case f is so 3 by 4, okay, plus 1 by 4 half.

(Refer Slide Time: 19:58)

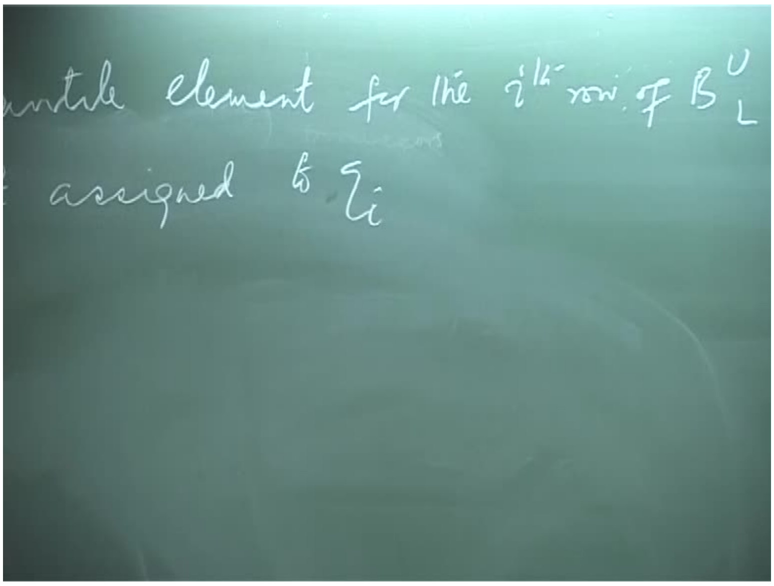


So, in that case that f is 1 by 4 means it will give you the one-fourth element first quarterly part, right, and f is 3 by 2 it will give you the hard quantiles. So, that is the definition of we are defining the quantile element.

(Refer Slide Time: 20:19)

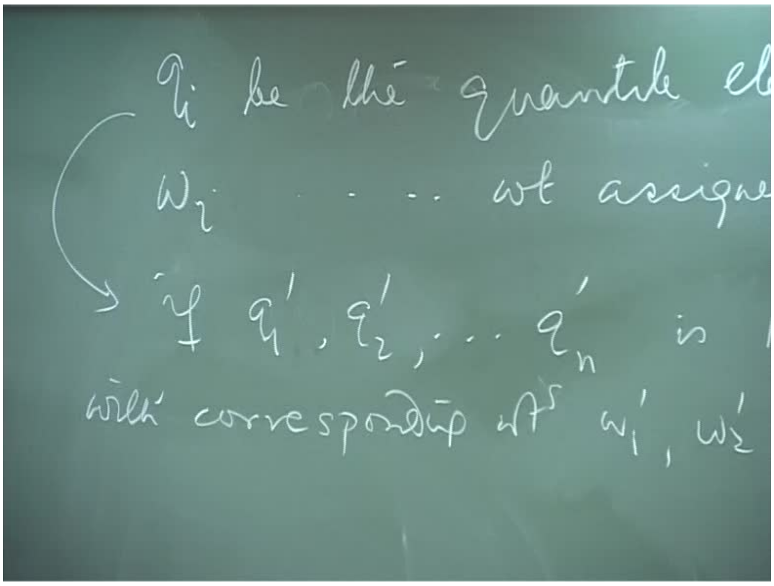


(Refer Slide Time: 20:28)



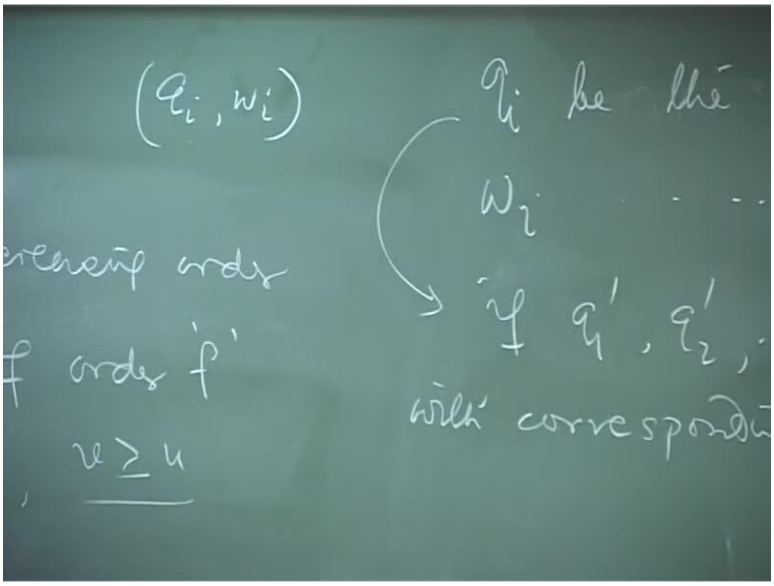
Now let q i be the quantile element for the i th row of the B U L. Let w i be the weight assigned to q i; you remember that q i nothing but U i minus l i plus 1.

(Refer Slide Time: 21:37)



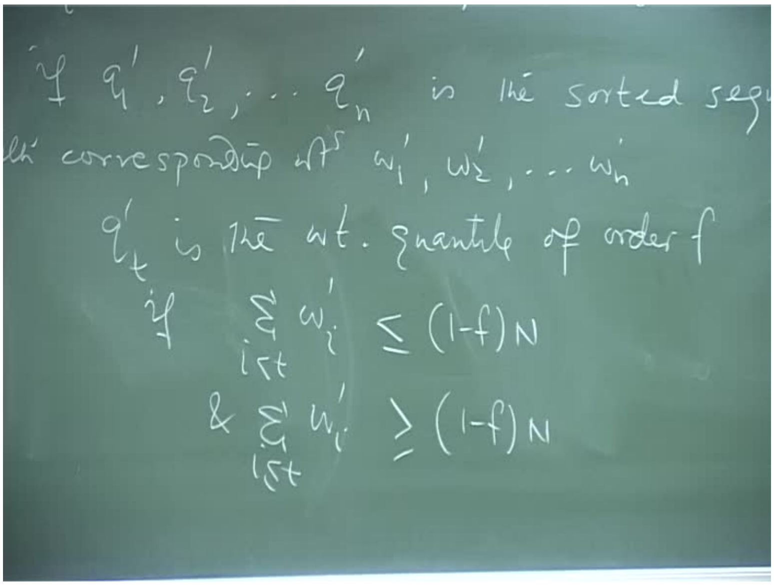
Then if q 1 dash, q 2 dash, q n dash is the sorted sequence of q 1, q 2, q n with corresponding weights w 1 dash, w 2 dash, w n dash.

(Refer Slide Time: 22:19)



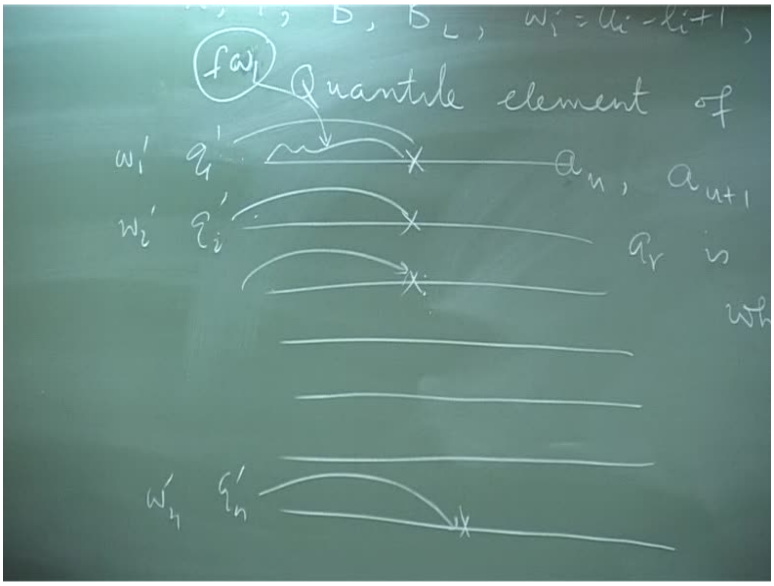
What I am telling that I have sorted, so it is a two-two poll you have q i, w i. Now you are sorting them with response to q's, so you will be getting q 1 dash, q 2 dash, q n dash and correspondingly weight also will be moved w 1 dash, w 2 hash, w n dash, agreed is it clear? No, just I have sorted q. Why i have sorted q? W is also attached to q, so w also changes position, nothing else.

(Refer Slide Time: 22:57)



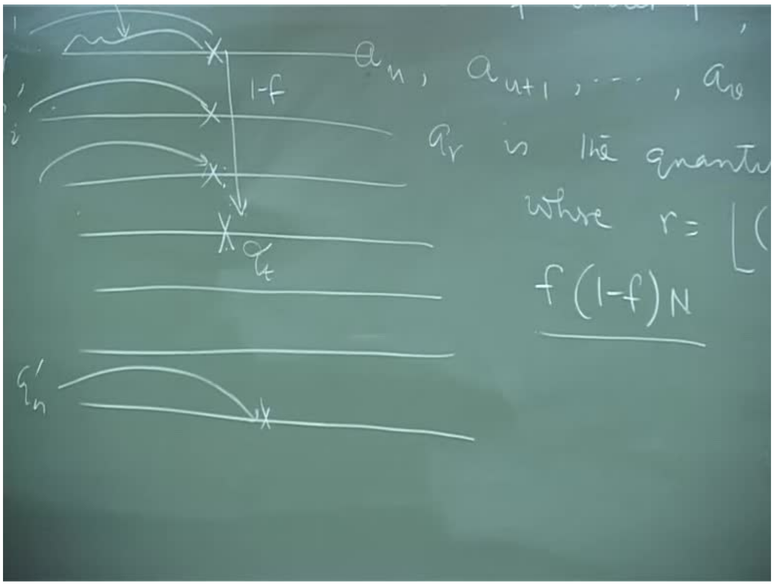
Now q t dash is the weighted quantile of order f if I sum this weight till t minus 1 if I find that it is less than equals to 1 minus f times of n. But at least the weight of the q 3 it is exceeding 1 minus f into n, then where q 3 dash is the, let me do this way, less than only, then q 3 dash is known as the weighted quantile of order. Now you think about this.

(Refer Slide Time: 24:28)



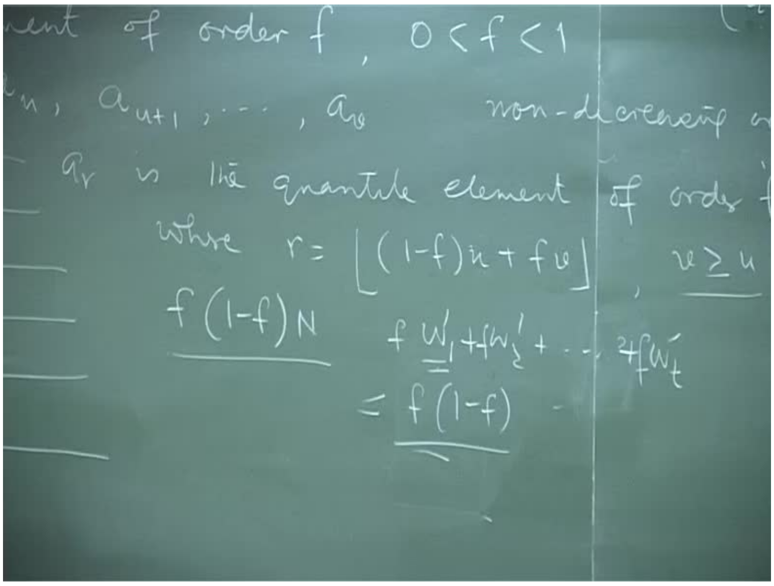
This is q 1 dash, q 2 dash, q n dash and weights you have w 1 dash, w 2 dash, w n dash, right. So, q n dash is this one, q 2 dash is also this one, q 3 dash is this one and q n dash is one. This is done of its own row this is the quantile, of its own row it is the quantile, agreed or not and if it is the quantile of order f that gives you the guarantee that there exists one-forth f into n th element smaller than this element, not n, f into w 1 elements smaller than this, agreed. Now similarly, f into w 2 elements will be smaller than less than or equals to this element, f into w 3 elements less than or equals to this elements and so on, agreed.

(Refer Slide Time: 25:54)



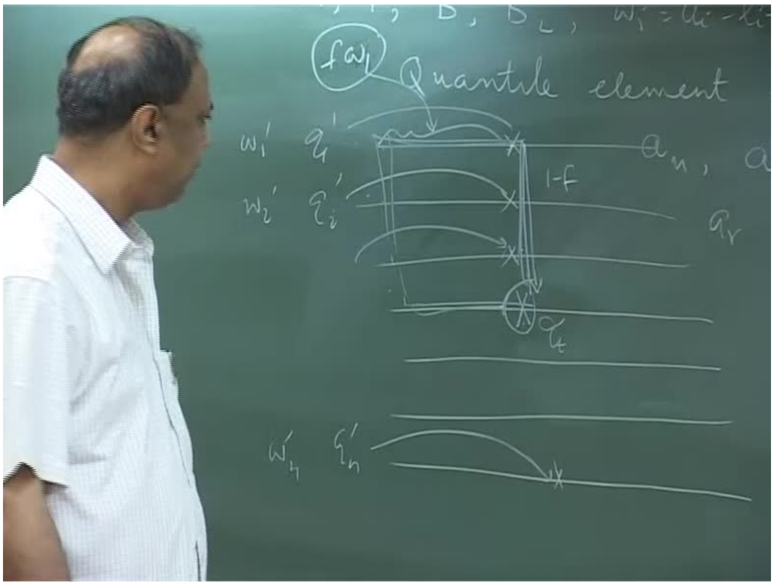
Now suppose this is your q t dash how many elements will be smaller than q t dash guaranteed. This is you are telling 1 minus f; so this is your 1 minus f, this side is the f, so you have at least that many elements will be this is 1 minus f and this is w 1, w 2, w 3, w 4, right.

(Refer Slide Time: 26:44)



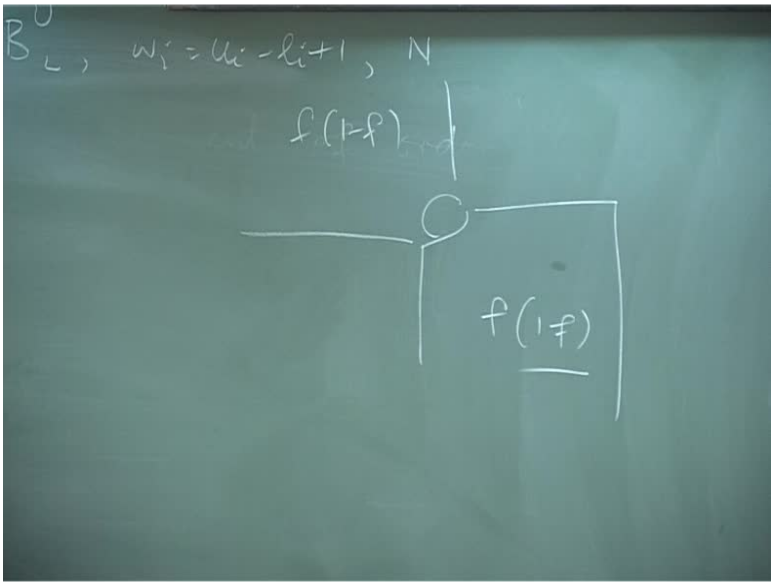
So, what I am doing here w 1 plus w 2 dash w t dash; w 1 dash gives you the guarantee of f time. This gives you the guarantee that f into w 1 dash, f into w 2 dash, f into w 3 dash, right, this is f of them f of them, they are less than; these are the sum, agreed. So, this gives you the guarantee that f into this side is 1 minus f of total n, so it is 1 minus f into size, right, agreed or not.

(Refer Slide Time: 27:36)



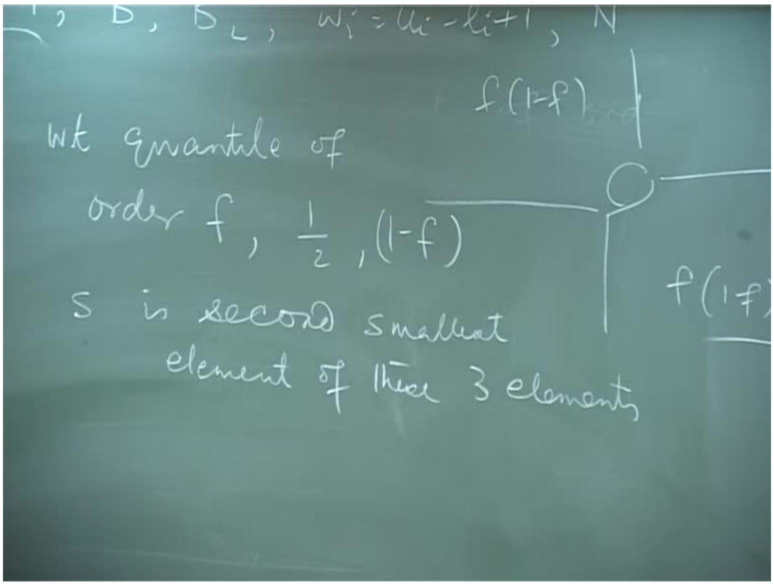
So, this gives you the guarantee that size of elements at least less than or equals to this element; yes, up to this we have done.

(Refer Slide Time: 27:57)



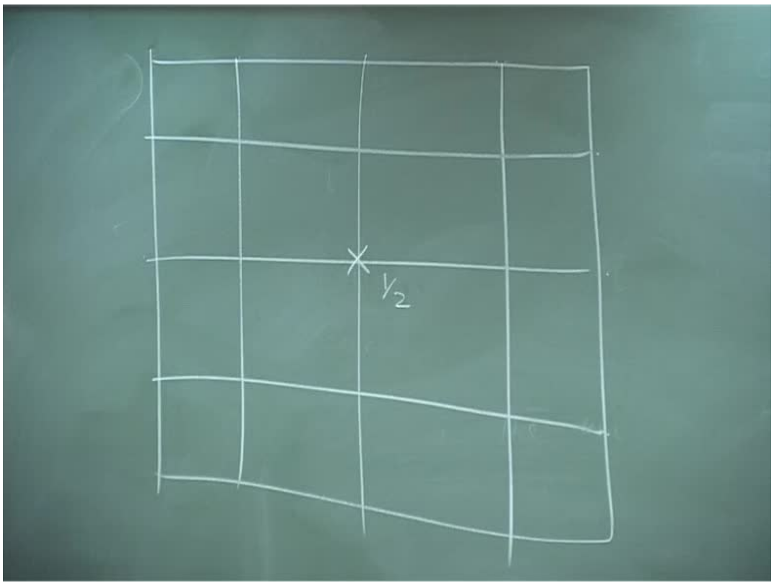
Similarly, you can see from the bottom side also this part; I have talked about this part that f into 1 minus f, similarly you can tell about this part also, right, because I am assuming that this is the element, so I can tell that f into 1 minus f, this side also you can define similarly.

(Refer Slide Time: 28:25)



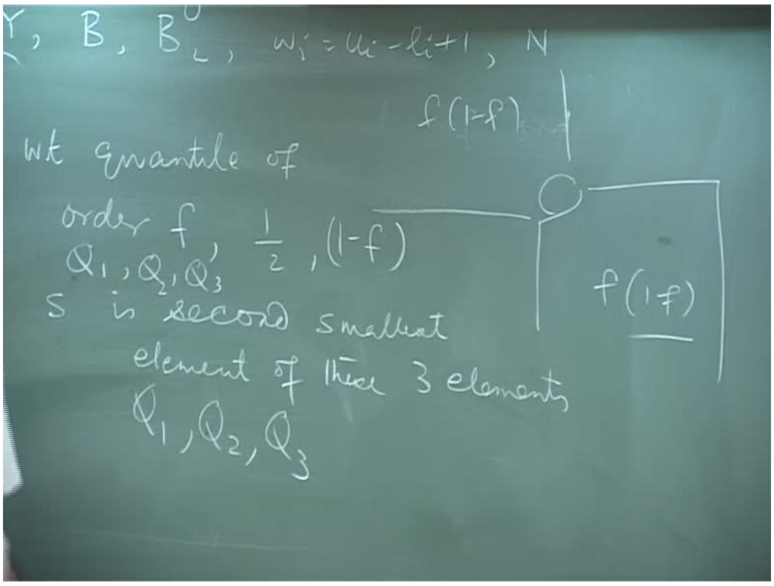
Now let us assume that you have weighted quantile of order f, half and one minus f, and s is the second largest element or second smallest element of this, right.

(Refer Slide Time: 29:16)



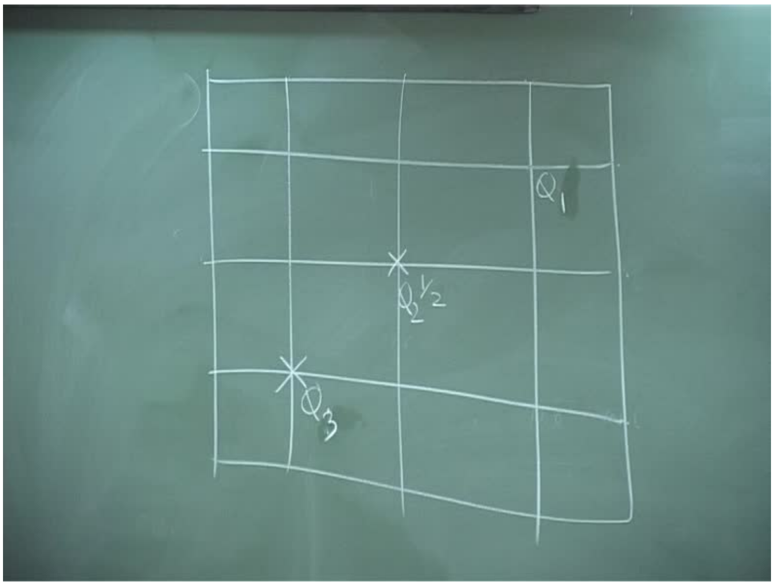
So, for simplicity we are assuming that B U L is of the structure and this is your order f order half, say let us name them first weighted quantile order f.

(Refer Slide Time: 29:55)



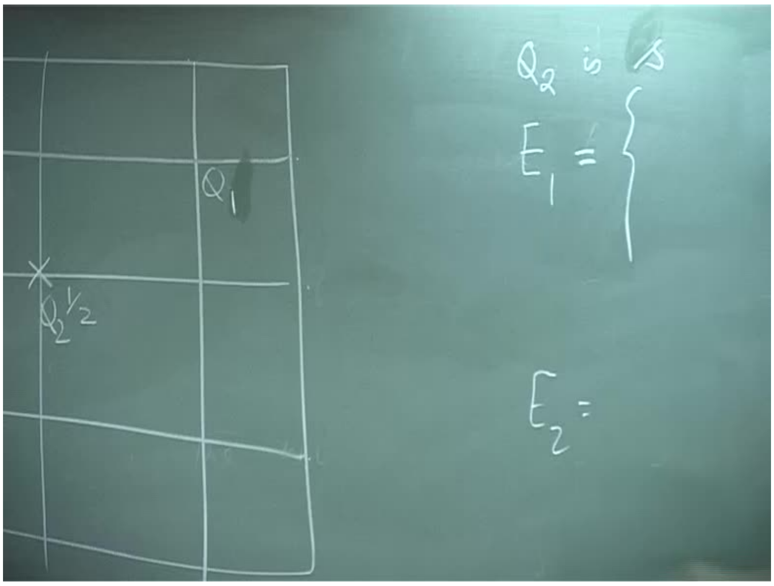
Let it be Q 1, Q 2 and Q 3, right, three weighted quantile one is name is q 1, q 2 and q 3 and s is the second largest element of this.

(Refer Slide Time: 30:18)



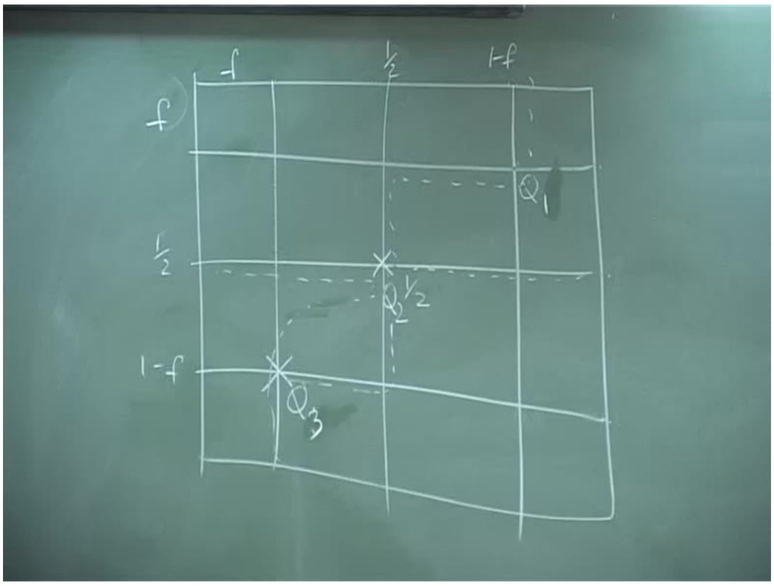
So, this is your Q 1, this is Q 2 and this is your Q 3, is it okay or is it reverse going? This is Q 3 or this is q 1 order f, f is 1 minus f; this should be Q 1, this should be Q 3, right, 1 minus n. Now one of them will be the second largest element, see what happens; suppose Q 2 is the second largest element, it means one of them would be the smallest element, right.

(Refer Slide Time: 31:13)



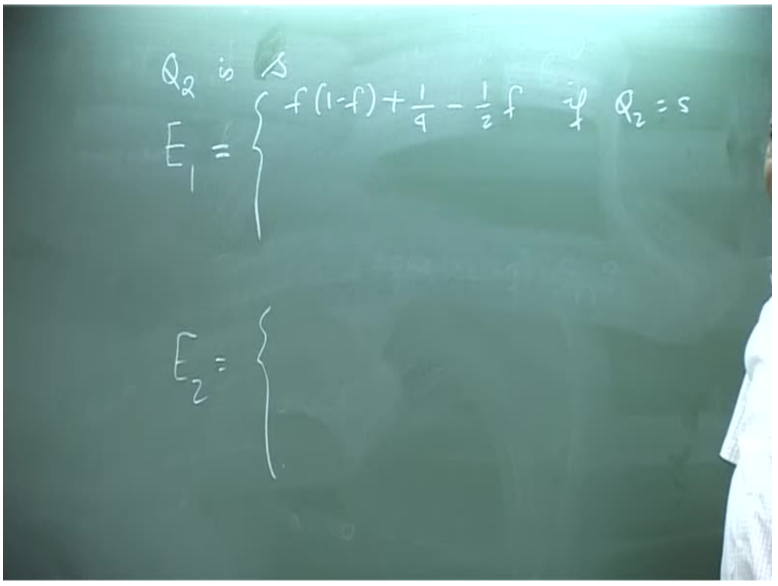
Let us assume that Q 2 is the smallest element second largest element. Q 2 is s. If Q 2 is s, what is the E 1 is the area or E 1 is the minimum number of elements which are smaller than Q 2 and smaller than s second largest element. And E 2 is the number of elements minimum number of elements which are larger than s.

(Refer Slide Time: 31:56)



Now if Q 2 is the second smallest element one of them is the smallest element, in that case suppose we assume that this is the smallest element then this gives you the guarantee that that many elements will be smaller than this because this is the smallest element, right, and this is the second large. So, this element will be smaller than this. So, all these elements will be smaller than q 2 agreed, and similarly, this is the largest element larger than this elements, all these elements are larger than, all these elements are larger than this elements; this is guaranteed. All these elements will be larger than this elements, all this elements will be smaller than this elements; this is guaranteed to you. Now what is the size of this? This is f, this is f, this is half, this is 1 minus f, this is f, this is 1 minus f.

(Refer Slide Time: 33:20)



Now the area is this one; that is f into 1 minus f plus 1 by 4 minus half f if Q 2 equals to s, agreed.

Student: One element smaller than Q 2? Are they along the half line or along this line?

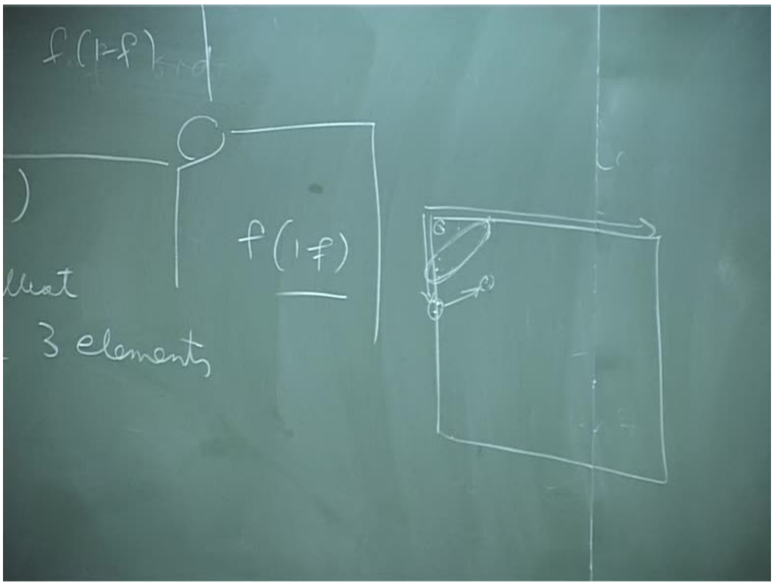
This line, all this elements will be smaller than this.

Student: Why sir? Why not this?

Which element? Bottom one? That is I do not know; may be but guaranteed, I am telling minimum number of elements. These elements can be smaller than this. So, i cannot tell that is true for any.

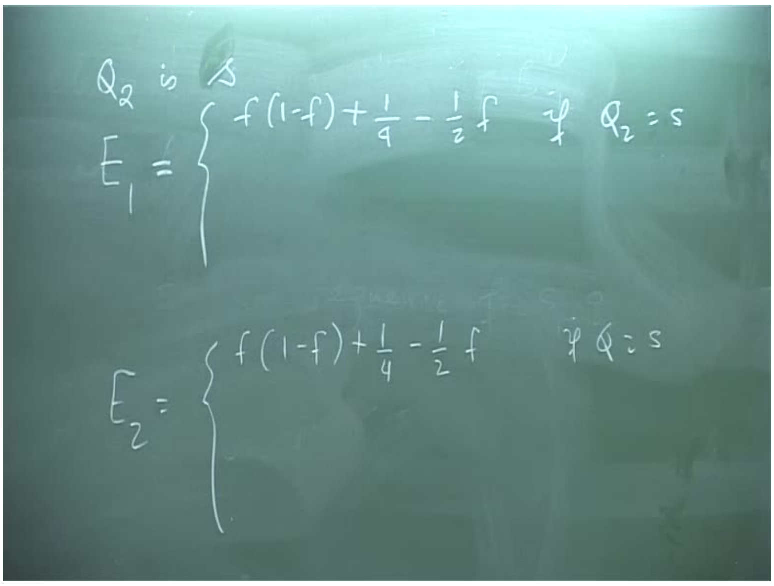
Student: Earlier when you took the second largest and third largest you showed that these elements will be definitely smaller than this.

(Refer Slide Time: 34:25)



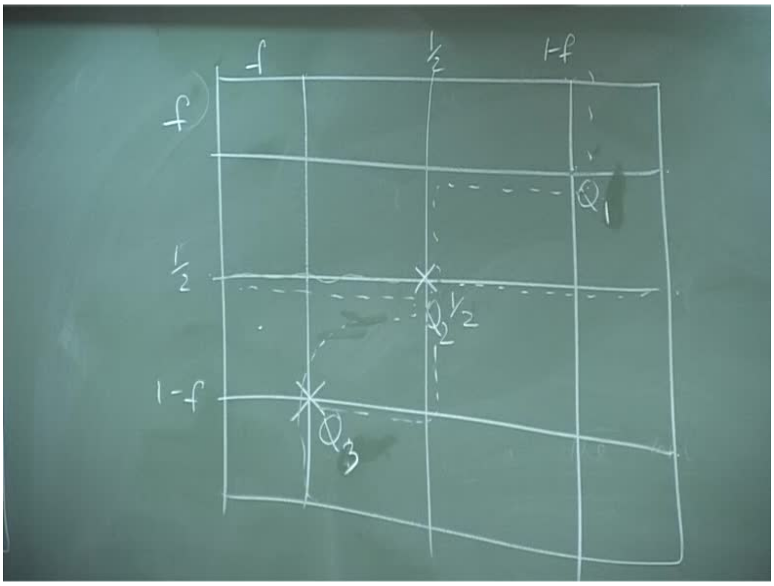
See, why I am considering the original matrix? I am telling this one, I am telling that this elements and this element is smaller than this. I never told that this element is smaller than this, right. There is no guarantee because guarantee is this is increasing, this is increasing, but there is no relationship between these two, agreed. Along the rows, along the columns relation maintains but diagonal relation is not there, it is not known, right, but this cannot be the fourth element, this cannot be the third element because this one two three, at least there exists three elements smaller than this; there exists three elements smaller than this. So, this cannot be a third element, but this can be a third element, because this element may be smaller than this element, right; that is the thing I told, agreed. So, is it okay?

(Refer Slide Time: 35:50)



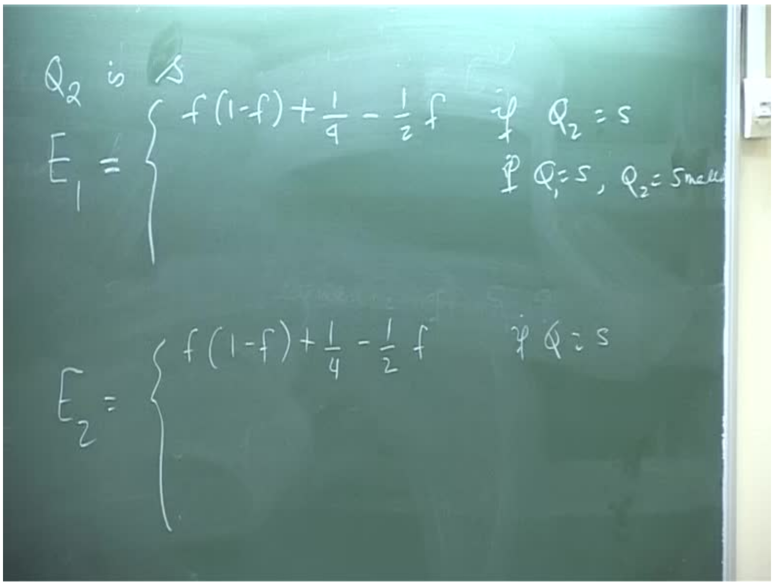
This Q 1 Q 2 is the small second largest element. Similarly, I can find out this side is f into 1 minus f plus 1 by 4 minus of f f if Q is s. Now if Q 2 is not s, suppose Q 2 is the first element or the third element, right.

(Refer Slide Time: 36:31)



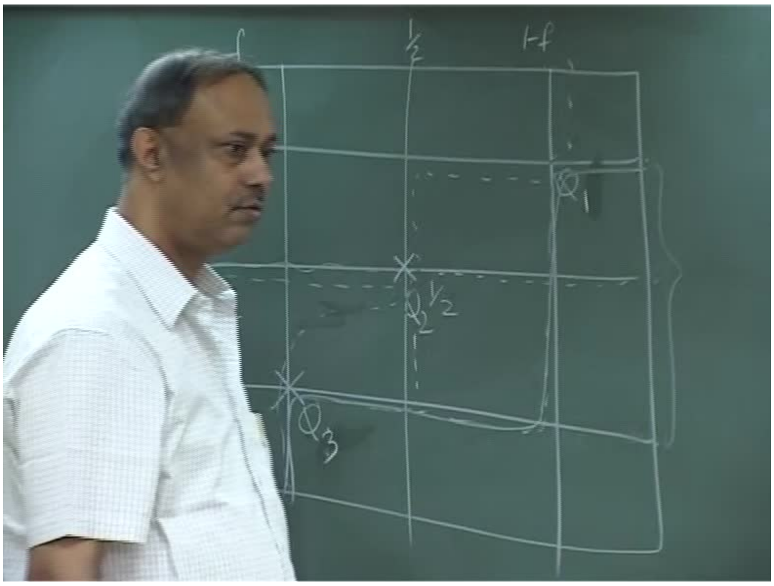
So, let us assume that Q 1 is the second smallest element and Q 2 is the first element, right. In that case, what it becomes?

(Refer Slide Time: 36:48)



If this is the second element, if Q 1 is s and Q 2 is smaller, in that case the area is same.

(Refer Slide Time: 37:06)

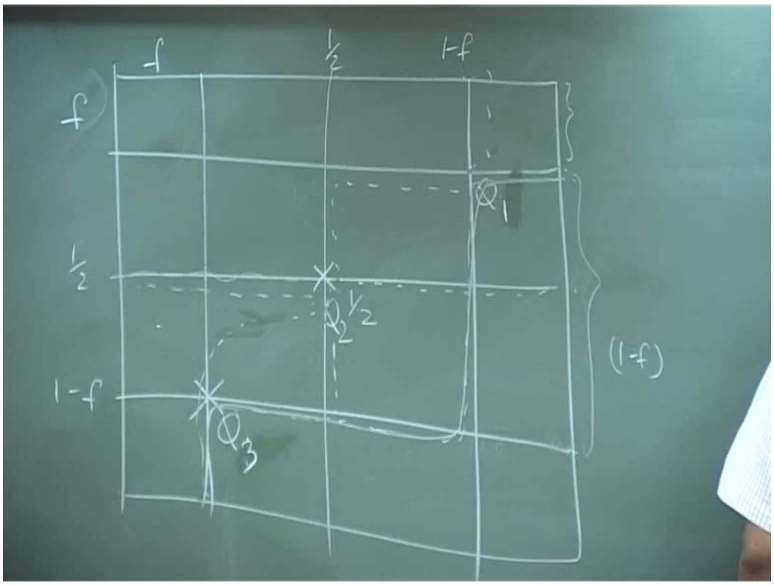


Area is same, agreed, but what about the other side? This is the second one; this is the largest one, what should be the area? Area will be this one, because this is the largest one, this is the second largest one. So, this gives you that these many elements would be larger than this element. So, that area is f into 1 minus f plus half f, is it half f? This is f, right, and this is 1 minus f, agreed or not, because this part is f. So, this is our 1 minus f. So, you got this area is f into 1 minus f, what about this area? This is f and this is also f.

Student: No sir, it is 1 minus f.

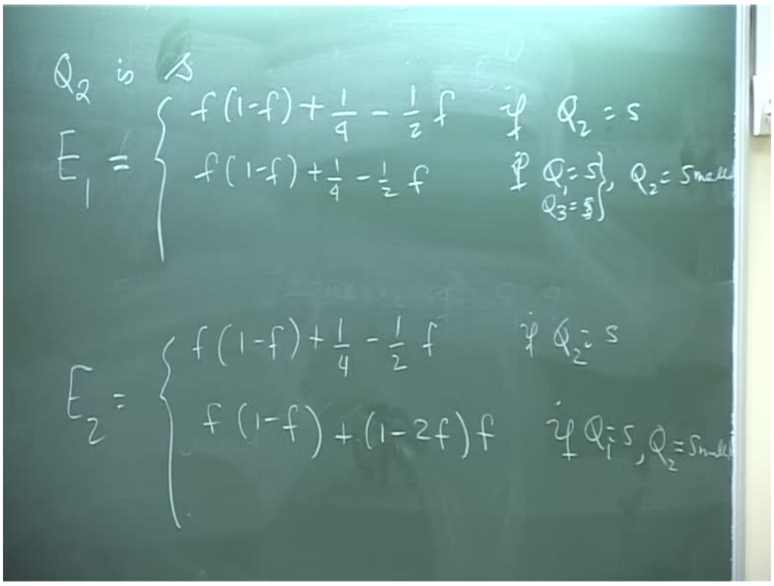
This is minus 2 s f into f.

(Refer Slide Time: 38:41)



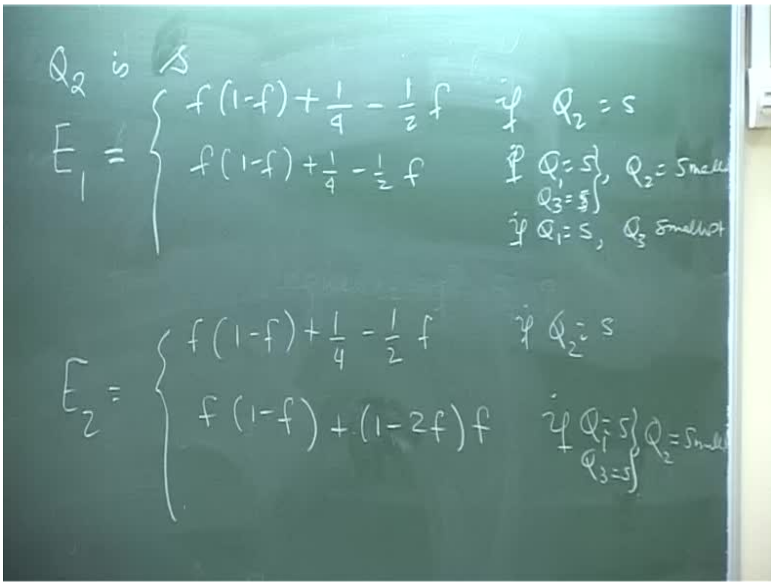
Okay, or I know this much 1 minus f minus f.

(Refer Slide Time: 38:55)



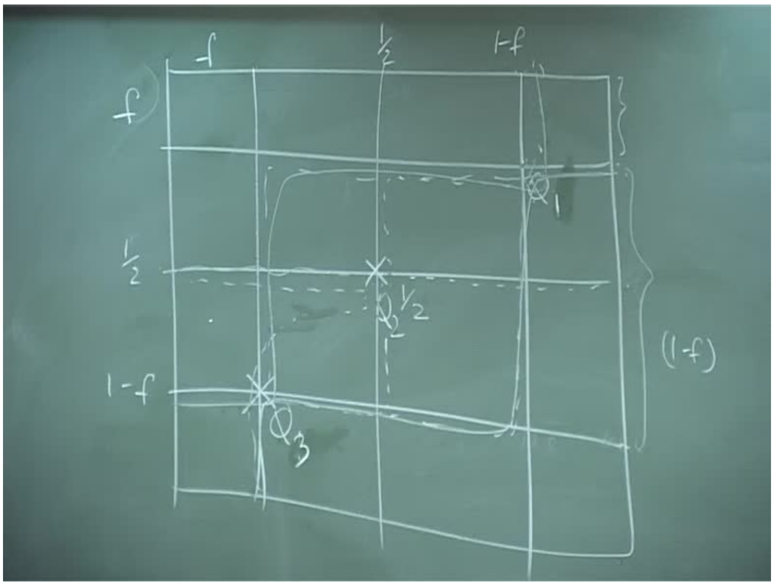
So, 1 minus twice f here into if Q 1 is s and Q 2 is 1. This side it is same f into 1 minus f plus 1 by 4 minus 1 by 2 f. Now if Q 3 is the smallest second largest element and Q 2 is the smallest, that is sent. So, I can write here q 3 is smallest for this case, say. What about here? If Q 3 is the second largest element instead of this, this is the second largest element and this is the largest element, so result is same.

(Refer Slide Time: 40:02)



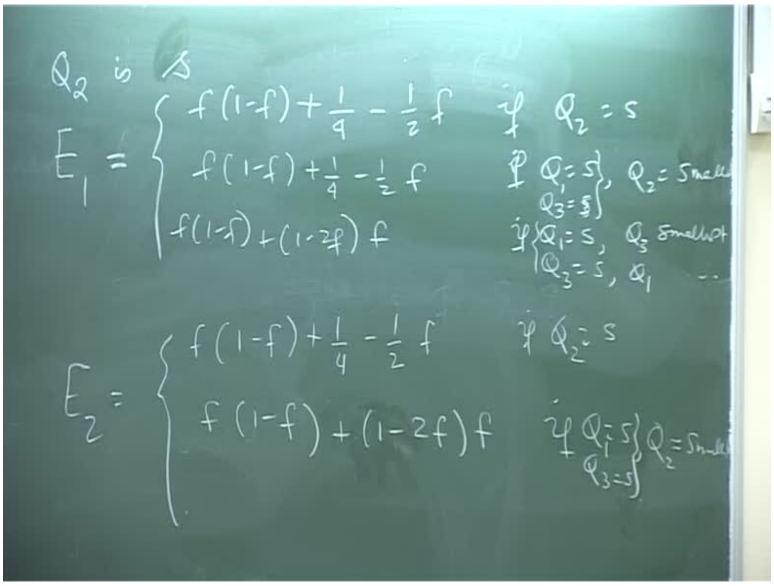
So, this is true for Q 3 is s also.

(Refer Slide Time: 40:11)



Now the thing is that Q 2 is the largest and Q 1 is the second. This is the smallest, this is the second, and this is the largest. What happens in that case?

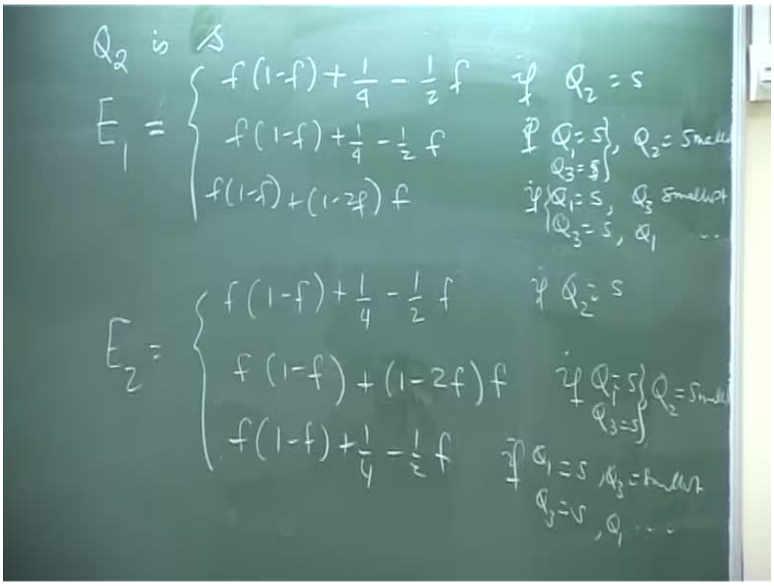
(Refer Slide Time: 40:23)



If Q 1 is second one and Q 3 is small, what is the value?

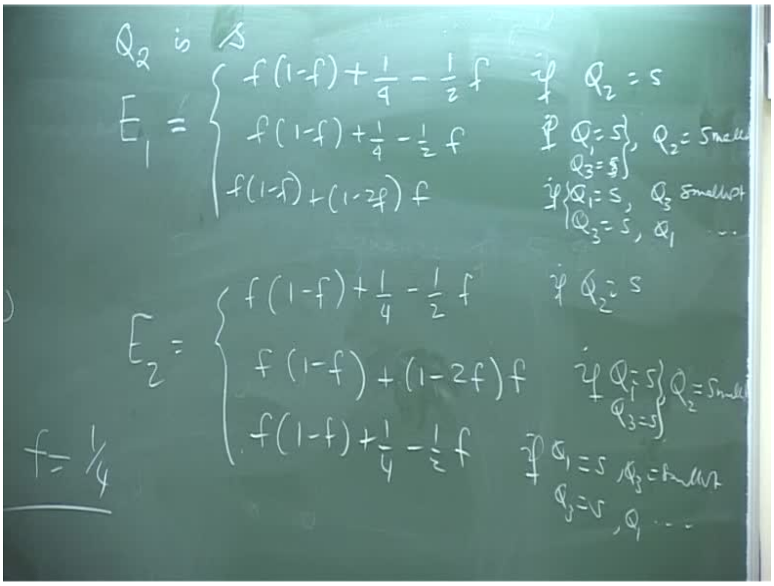
If this is the second one, this is the smallest, then size is guaranteed part is this part, agreed, which is nothing but f into 1 minus f plus q minus 2 f into f, and this is true for the other case also Q 1, Q 3 for second and the Q 1 is smallest, agreed. Now what about here?

(Refer Slide Time: 41:15)



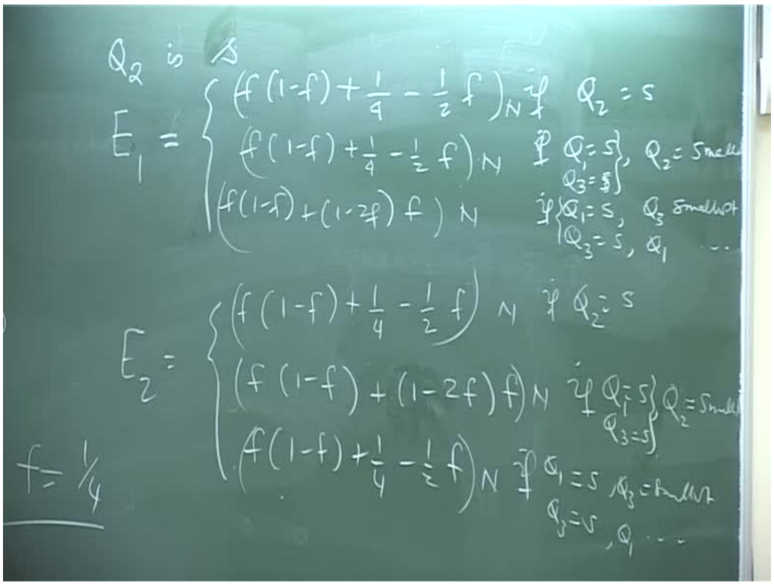
This becomes this one, so this becomes here f into 1 minus f plus 1 by 4 minus half f if Q 1 Q 3 is s and Q 3 is smaller than Q 1 is s, right. Now your aim is to maximize these two areas maximize E 1 and E 2, right. If I can maximize that one that will give you, so for what value of f you get E 1 and E 2 maximum; that means E 1 and E 2 will give you how many elements you can describe at each solution.

(Refer Slide Time: 42:12)



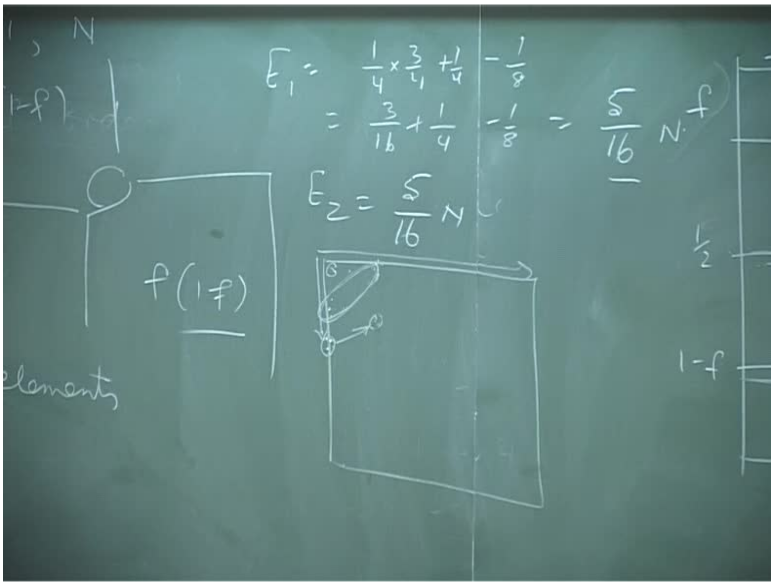
So, if I just simply take the d by derivative of this you will be finding that f is maximum at 1 by 4; this function E 1 and E 2 will be maximum when f equals to 1 by 4. E 1 and E 2 is maximum with f; otherwise, if you increase f you may find E 1 is increasing, you will find E 2 is increasing, so that will not help you. You have to optimize in such a way that you get the guaranteed that that much difficulty part is there. So, f is 1 by 4 and if f is 1 by 4, then what is the value of E 1?

(Refer Slide Time: 43:00)



You name but this contains all of them are having one multiplication factor, what is that, right, total size, this is the factor. So, total size is there. If f equals to 1 by 4 what happens?

(Refer Slide Time: 43:24)



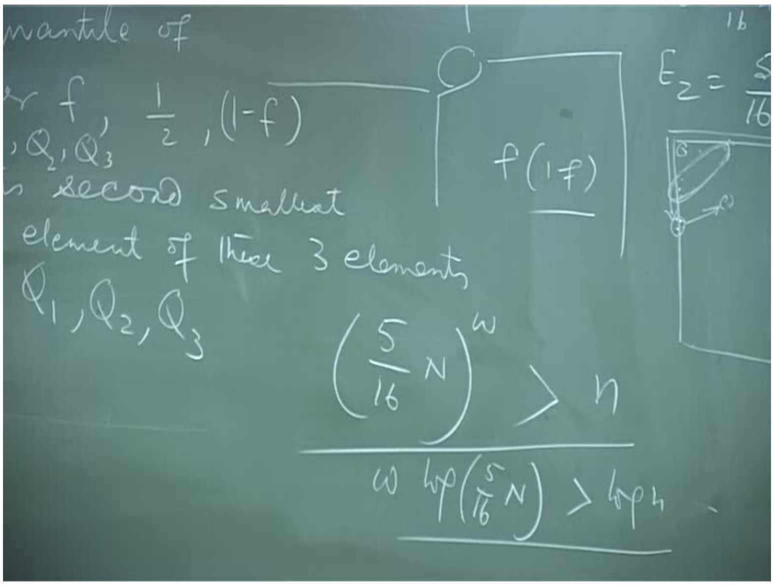
The value is what 1 by 4, 3 by 4 plus 1 by 4 minus 1 by 8, 3 by 16 plus 1 by 4 minus 1 by 8, what is the value?

Student: 5 by 16.

5 by 16, so you can find out for everywhere it will become 5 by 16 and 5 by 16. So, it gives you that if I considered the second largest element of this three quarters, because f is 1 by 4, so it is first quantile, second quantile and third quantile; out of that if I use the second largest element or second smallest element to discrete that gives you the guarantee that at least 5 by 16 n elements will be discarded at each iteration.

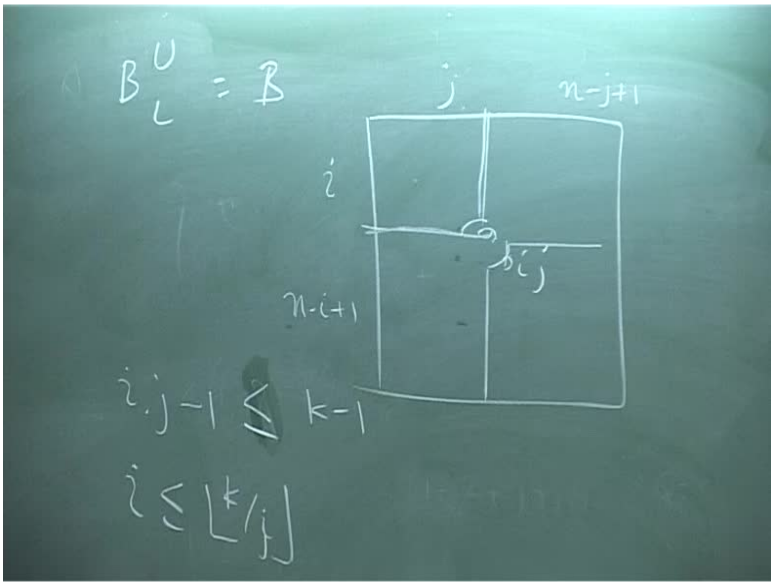
In the initial case or if I remember the first finding the case element it was having that 4 by 16; that is 25 percent. If you remember it is 25 percent you can discard at each iteration here; instead of 25 percent you discard it 5 by 16 elements, right. So, at each iteration you will be able to discard 5 by 16 elements of b from your possible sub-elements. Now if you have n elements and n processing elements you just sort it and find the k th element which takes more long in time; we will just sort it, sorting takes long in time using n th processor.

(Refer Slide Time: 45:22)



So, this iteration 5 by 16 N to the power w, it will be iterated till it is greater than n. If I find it is less than equals to n I stop it. So, w is the number of iterations; I will iterate it, every time it is reducing by 5 by 16 times, right, and till I find it is equals to greater than equals to n, I will continue. So, I can find out this iteration should also be what, how many times? So, you just take w log of 5 by 16 N is greater than equals to log n. So, you can find out find out what is the value of w, right; w is also factor of initially capital N is your order k, there are incomplete order. So, you can find out this iteration would be of the order of log k. So, I can find out the number of iterations you need to reduce the size of the B U L of the order n, so that you can use that n th processors to sort this n elements and find the k th element; it means constant time.

(Refer Slide Time: 47:16)



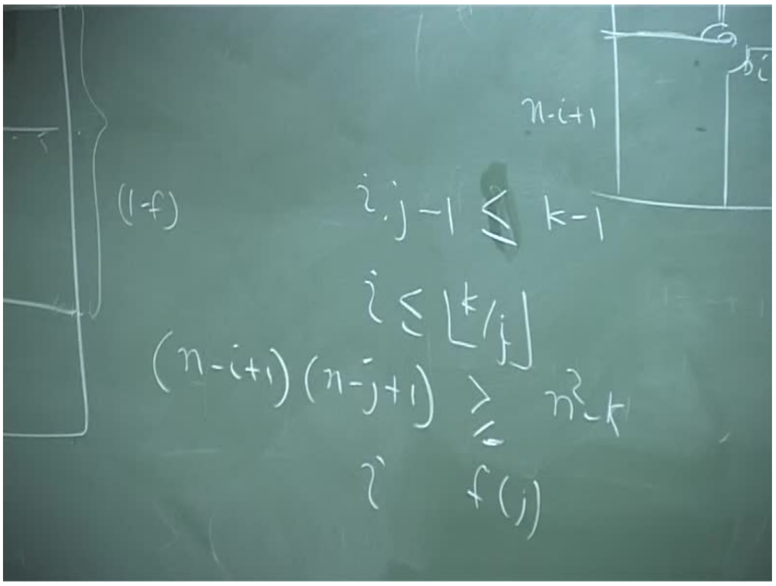
Again so next part is that initially how I can define B U L. I can start with B U L is your B, but as i discussed with you that depending upon the value k, B U L can be defined because as I told that for first smaller value of k I should not have defined B U L equals to B, right. So, this can be obtained from this one. Suppose you have this and this is your b i j, b i j ij th elements of the matrix x plus y. b i j is greater than equals to at least i into j minus 1 elements. This is i, this is j, all this elements will be smaller than this term, right. So, the number of elements is I into j minus 1, agreed. This is given to you. Now b i j is the k th element; that gives you that exact k minus 1 element will be equals to i with i j minus 1 in this, yes.

Similarly, this side you have if it is j, this side is n minus j plus 1, this side n minus i plus 1. So, b i j is the k th element, then exactly equals this n minus j plus 1 equal this, that much be n square minus k th element, yes or no. This element is what? Total number of elements n minus j plus 1, agreed, sign goes to n minus 1 plus 1. So, this area is this into this n minus i plus 1 and n minus j plus 1 and if it is the k th element, then the total element is n square, so n square minus j will be this size, yes. Now b i j can be, i am not talking that this is the k th element, b i j can be k th element if tell me what? b i j can be the k th element if?

Student: Less than equal to.

Less than equals to, this is one condition you can easily obtain which tells you i is less than equals to k by j; that is guaranteed, alright. I could have taken it apart will not give you the guaranteed thing throughout gives you the guarantee.

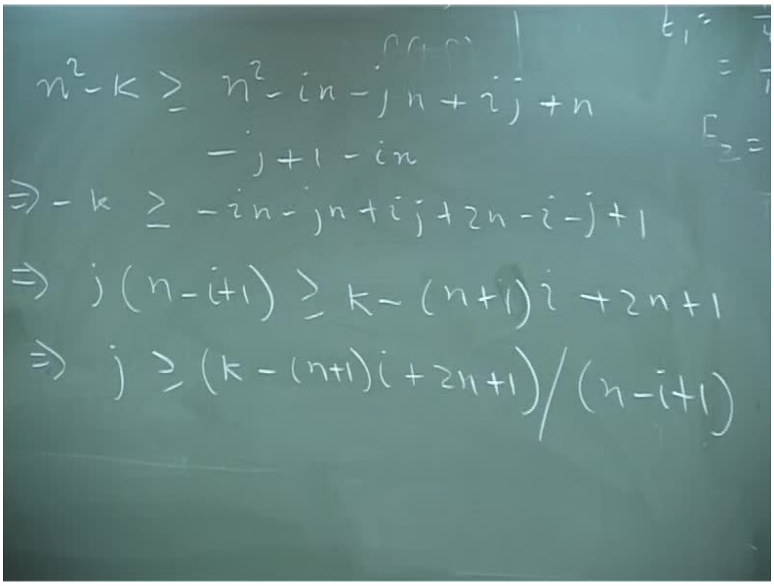
(Refer Slide Time: 50:52)



Similarly, I can obtain the n minus i plus 1 into n minus j plus 1 n square minus k, what should be the symbol here? Greater than equal, is it greater than equal I will write, right, and from this also you will be able to find out the functioning terms in terms of j, yes or no? It is just not possible.

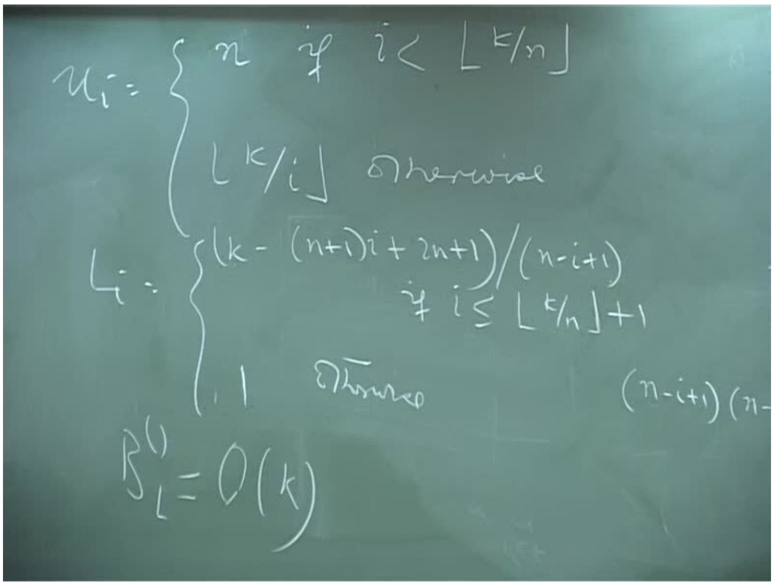
Student: In terms of n and j.

(Refer Slide Time: 51:35)



j n plus i dot j plus n minus j plus 1 minus i n; this gives you minus k greater than equals to minus i n minus j n plus i j plus 2 n minus i minus j plus 1. This gives you j n minus i plus 1 greater than equals to k minus n plus i i plus 2 n plus 1 which is equal to j greater than equals to k n plus 1 i plus 2 n plus 1 n minus i plus one. So, this is after calculation you will be getting this.

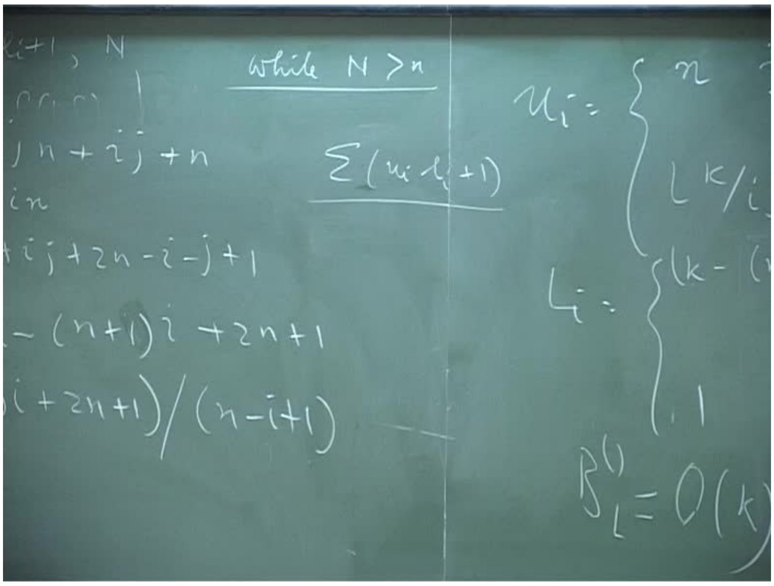
(Refer Slide Time: 52:39)



So, if you know this then i can define your U i is n if i is less than k by n and k by i otherwise. L i is your k minus n plus 1 i plus 2 n plus 1 n minus i plus 1 if i less than equals to k plus 1 and 1 otherwise, right. Now you know for different value of i you can get U i and L i now and for you at this point is your because lot of arithmetic will be coming. So, initially we have seen that B U L is order k through some how many numbers. So, we have shown that B U L is order k, right. So, if k is very small, U L size is very small, you just sort it and get it. If k is order n square then you iterate it and we have seen that after log k iterations the size will be order n and that is, right.

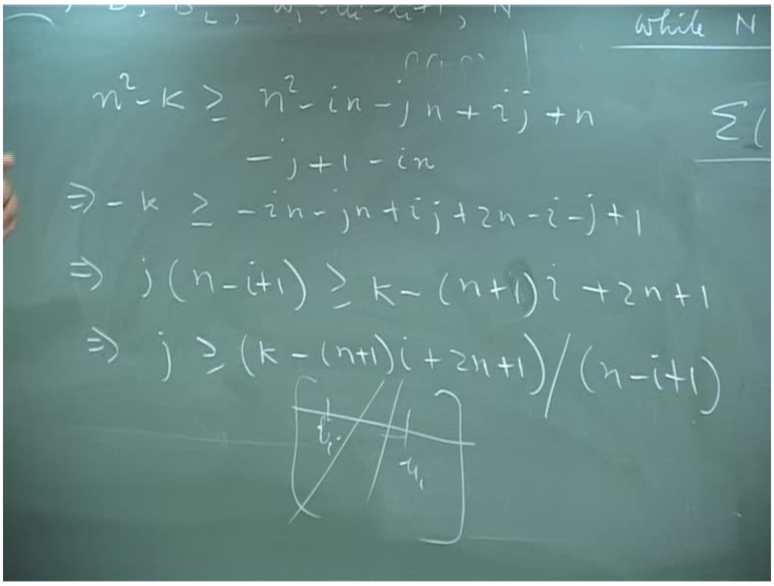
Now, think about the whole process; how much time you need, if you have n processors? First what we have decided that you have x and y, use the starting algorithms which takes order log n time to sort using n process, right. Then second step is what? You have to initialize B U L that initialization B U L everybody finds is I; it takes constant amount of time, right. So, initialization B U L takes what? Constant amount of time, then you have to know what is the value of w I, that takes constant amount of time because every processors find q i minus l i plus 1. Then you need to know, what is the value of n, because you have to check whether N is smaller than or equals to small n or not, agreed.

(Refer Slide Time: 55:32)



So, you have to find out capital N, you need to know summation over u i minus l i plus 1, you have n processors, n summations log n time, right, not more than, agreed. So, you can find out the capital N, then there is a for loop while capital N is greater than small n, right. You have to find out what? First quantile, second quantile and third quantile using n processors, right, yes which is how much time? Log n, then second largest element is compared to the three element get over constant time, then you divide into the three parts, right; that element second largest element has to be brought to first step to all the processors which takes log n time.

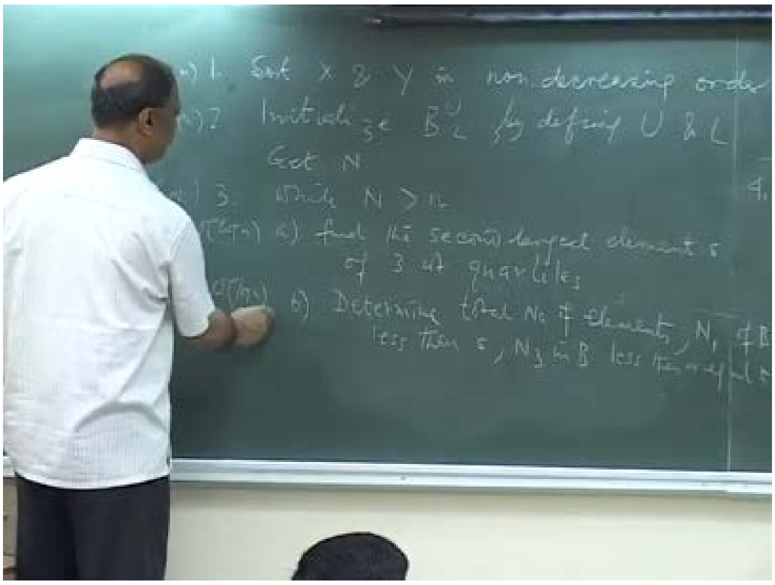
(Refer Slide Time: 57:00)



Each processors are always array, you have l i, you have u i; he finds out that the number of elements smaller than number of elements smaller than greater than, that he is finding, right. So, that is since it is in sorted order binary search worst case log n time he needs, because n elements may be there; he finds it in log n time the number of elements smaller than number of elements larger than, right. Now the number of elements smaller than you has to add them to find out that size if it is greater than k and other things.

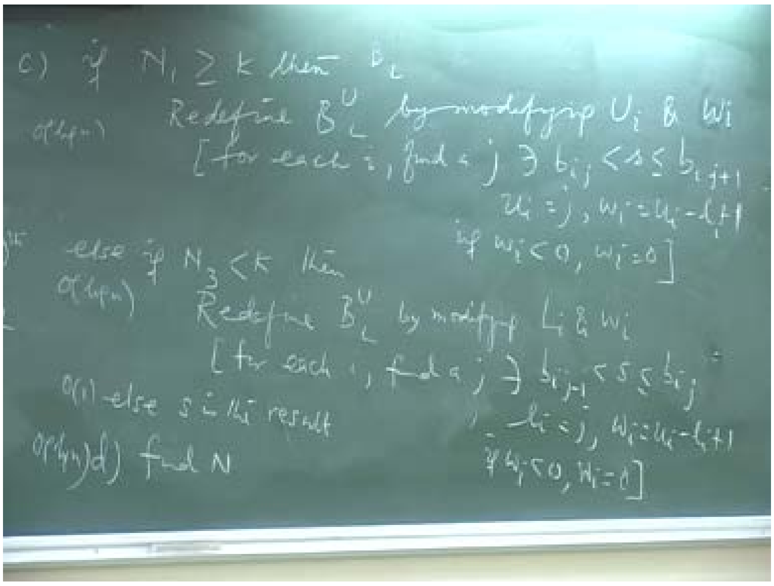
We will be assuming that it is the searching zone; otherwise searching zone, otherwise this is the element. Yes, add this takes log n time. Everything can be done log n time. So, the type of decimal becomes log N into log k, log k time iterations. Now after that also if you do not get the k th elements, the size is reduced to n sort it and to log n time find the k th element, right. So, now can you write your algorithm or no, you need I should write the algorithm. Can you write?

(Refer Slide Time: 58:40)

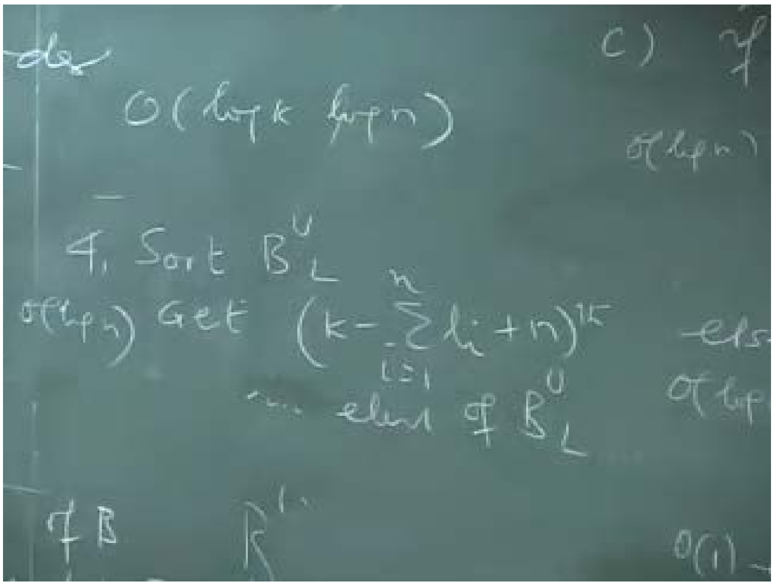


One is sort X and Y in non-decreasing order. Initialize B U L by defining the U and L get N while N is greater than n, a) Find the second largest element s.

(Refer Slide Time: 01:01:03)



(Refer Slide Time: 01:03:54)



So, this takes order log n time, this takes order log n time, this takes order log k time iterations, order log n time, order log n time, because addition is coming into that. This takes order log n time binary research, this is order log n time, this is constant time, this is order log n time, this is order log n time. So, the time complexity is order log k into log n times.

This is matrix, okay next class, I will cover matrix and thank you all.