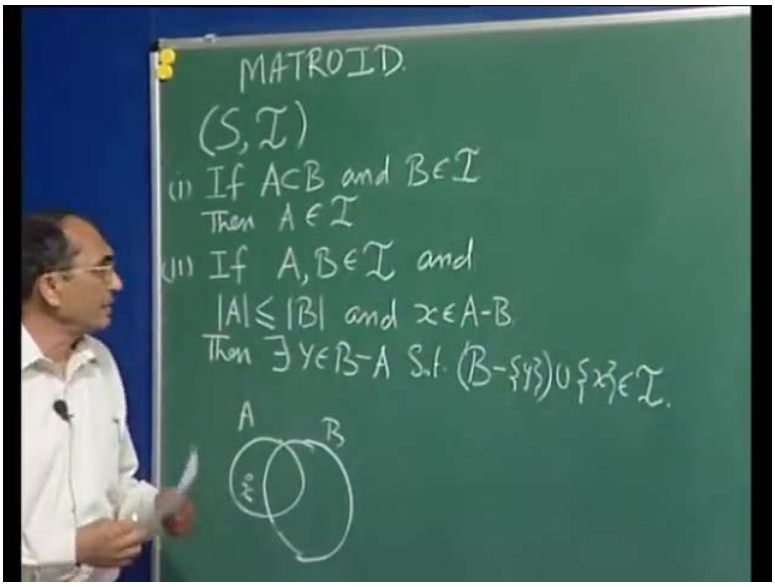
Computer Algorithms - 2
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 5
Matriods

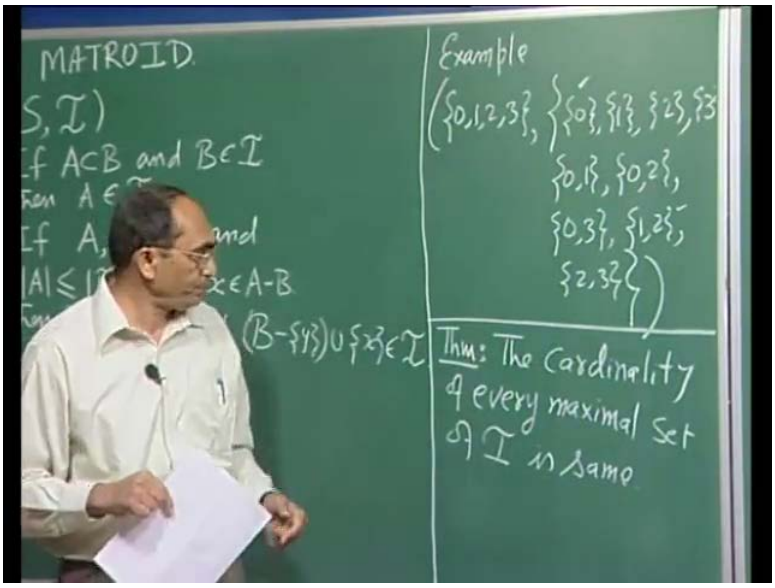
Today, we will discuss a structure called Matriod, and we will discuss an algorithm related to it. Matriods provide in fraction for many problems in graph theory and in some other domains.

(Refer Slide Time: 00:34)



A matriod is a couple where S is a finite set of a elements and this script I is a collection of some of the subsets of S, which satisfy certain conditions. Condition 1 is that, if A is a subset of B and B belongs to I, then A also belongs to I. The second condition, that this collection should satisfy is that, if A and B both are members of I and the cardinality of A is no greater than the cardinality of B. And there is some element x in A minus B that is to say, if visualize this way, then there is an element y in B minus A, such that B minus y union x also belongs to I. These two conditions must be satisfied by this family of subsets of S.

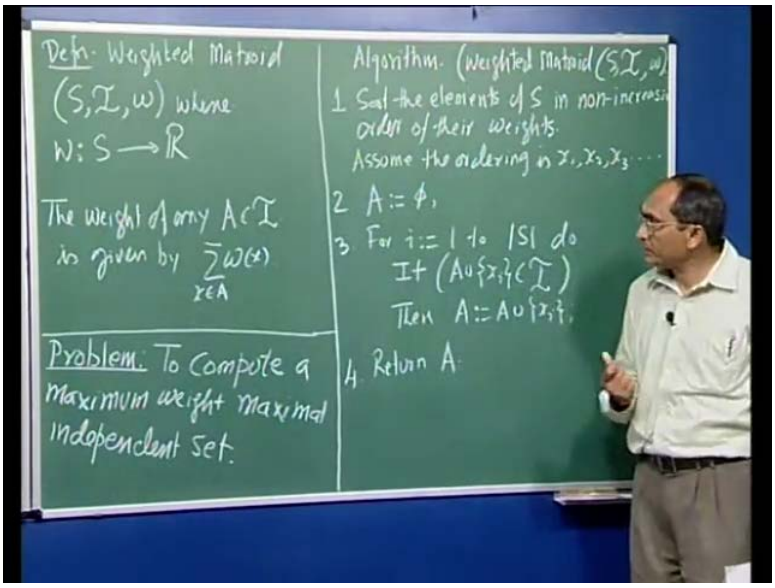
(Refer Slide Time: 02:55)



Here is a small example so I have set S containing 4 elements and the script I, the family of subsets contain the singleton 0, 1, 2 and 3, this five additional sets containing 2 elements. Now, if you consider this set and this set, if take this as A and this as B then 0 belongs to A minus B then there is an element in D namely, 2 such that, B minus 2 union 0 namely 0, 1 is also a member of this collection. The subset property is quite trivially satisfied in this case so this one can verify that this is an example of a matriod.

Now, there is one theorem, which I will leave for you to do verify yourself and the theorem says that, the cardinality of every maximal set of I is same. By maximal set, we mean those sets of I, which are not properly contained in any other set of I and the claim is that, all sets have same cardinality in a matriod. One more thing, we will call the members of I independent sets for some historical reason. So, in our example, all the maximal sets are 0 1, 0 2, 0 3, 1 2 and 2 3, they all of course, have cardinality 2.

(Refer Slide Time: 06:10)



Now, we define a weighted matriod, a weighted matriod is a three couple, S I and a weight function where, weight function assigns weight to every members of S where, weight is a function from S to real numbers. We will accept both positive and negative numbers now then the weight of any member A of I is given by some w of x, x belonging to A.

It is simply the sum of the weights of the elements of independent set A now, our problem that we want to address today, is to determine one of the maximal independent sets, one of the sets with largest cardinality in I, which has maximum weight. So, the problem that we want to address is to compute a maximum weight maximal independent set notice that, this maximal refers to the cardinality of the set and this refers to the weight.

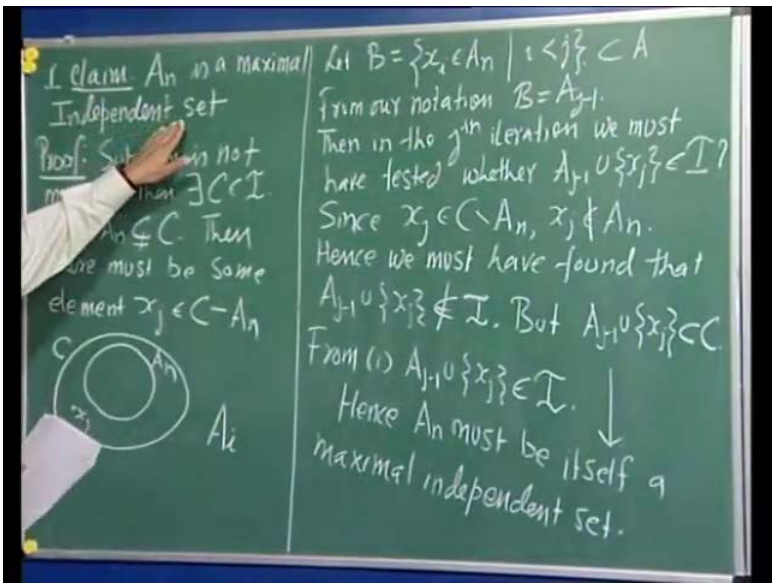
So now, we are going to give a simple method and I will not call it an algorithm yet, will verify the correctness of a method and then later on will look at an example by taking an application of a matriod method to a graph form. So, let us take the algorithm, which seems pretty trivial, a greedy algorithm to compute such a set. So, you are given a weighted matriod, in step 1 we will sort the elements of S in non increasing order of their weights.

For our convenience, I will assume that the ordering is x 1, x 2, x 3 where, the weight of x 1 is greater than equal to weight of x 2, weight of x 2 is greater than equal to weight of

x 3 and so on. In the second step, we take an empty set and now, we iterate in the following fashion, so all we are doing is, we start with an empty set and starting from the highest weight element, we put it in set and check, whether it is still an independent set because we are interested in independent set.

If it is still independent then we update A by putting that element and then move on to the next element and finally, we return the final output A. Now, notice that, in this so called algorithm, there is a step which checks whether certain set is independent or not. Now, in general, a matriod may not be given to you in explicit manner and hence, one does not know, whether such a step is feasible to evaluate and that is why, this is not quite in algorithm in classical sense because each step must be feasible. But, if it is possible to evaluate this, to check the membership of this entity then it becomes a meaningful algorithm. So, next thing we are going to do is, try to prove that this indeed computes our object namely, the maximum weight maximal independent set.

(Refer Slide Time: 13:46)



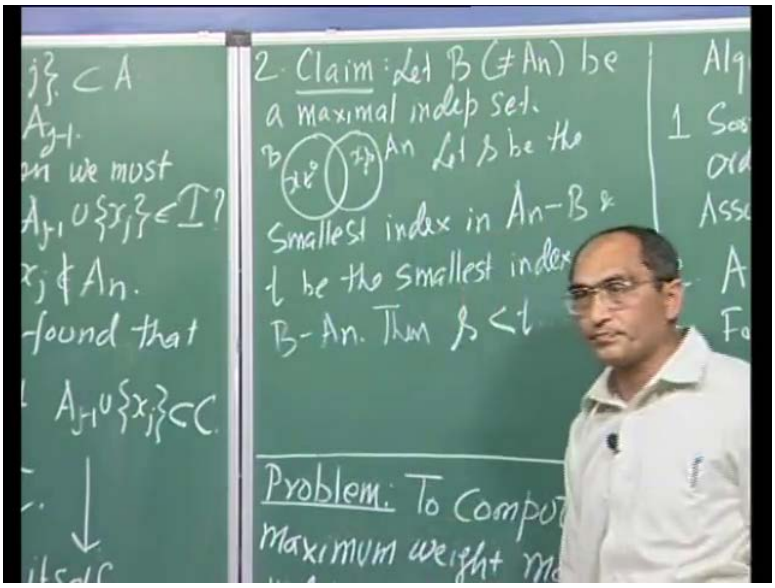
So, let us start with first claim, in order to define this claims, I am going to use certain notation, I will denote the value of A after i th iteration by A i, A subscript i. So, the final output is A n will say, A n which is the value of A return in final step so first claim is that A n is A maximal independent set. It is quite obviously, an independent set because in each step, we have checked whether current A belongs to script I. So, it is obviously, an independent set, but it is not clear, whether it is A maximal independent set.

So, let us start prove this suppose, A n is not a maximal independent set suppose, A n is not maximal then there exist a set C in I such that, A n is strictly contained in C, this means that, it is contained in C but it is not equal to C. Now, since this is strictly larger than A n then there must be some element x j in C minus A n, in the difference set there must be an element x j. So, let us just put the picture here, this is A n and this let us just draw this way, this is C and here is an x j.

Now, let us look at all the elements of A n with index less than j, let B be the elements of so B is a subset of A. Now, what we have decided to do is, call the set A after i th iteration by A i, A subscript i then notice that this set from our notation B is precisely A j minus 1. Because, from A n we have deleted those elements, which had index greater than j, greater than or equal to j of course, x j is not in A so does not matter. Then in the j th iteration, we must have tested whether A j minus 1 union x j belongs to I, since x j belongs to C minus A n, is not a member of A n.

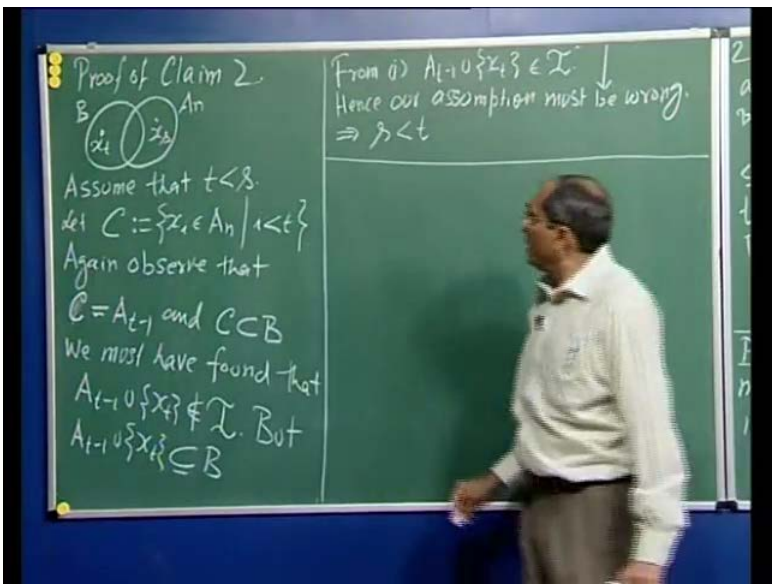
Hence, we must have found that, A j minus 1 union x j does not belong to I, this is the exits does not belong to A n hence, we must have found that, this set is not in I. Now, on the other hand but notice that A j minus 1 is a subset of A n hence, it is a subset of C and x j is also a member of C. So, we conclude that, A j minus 1 union x j is a subset of C now, from matriod condition 1, from 1 A j minus 1 union x j must belong to I. So, we find a conflict here, a contradiction hence, it is not possible to find any element x j in this, which implies that A n must be itself a maximal independent this is what, we wanted to establish that, at least it is a maximal independent set.

(Refer Slide Time: 21:18)



Next, we want to prove another useful claim and the claim is the following, let B not the same as A n be a maximal independent set. So, I have some another maximal independent set B, let us just draw the picture here of course, both of these subsets are same cardinality, because we have already claim that, all maximal independent sets have same cardinality. Let s be the index, the smallest index of the elements in A n minus B, let s be the smallest index in A n minus B and t be the smallest index in B minus A n then we want to prove then we want to show that, s is less than t.

(Refer Slide Time: 23:25)

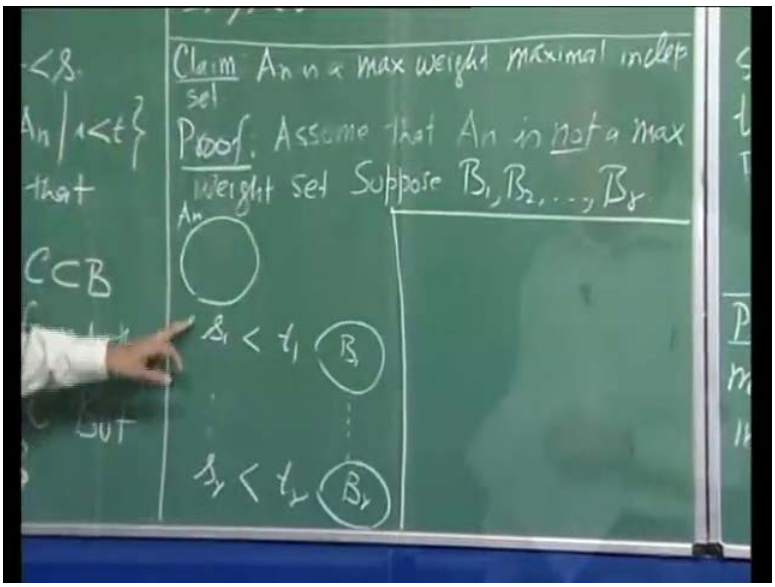


The proof of claim 2 so what we have is set B and set A n here, we have x t where, t is the smallest index in this region and x s where, s is the smallest index in this region. Assume the contrary, assume that, t is actually less than s and let us arrive at some kind of contradiction. Now, let C be the set of elements x i in A n such that, i is less than t so we are taking all the elements of A n with index less than t obviously, t is not in A n so it will always only up to t.

Now, clearly as you noticed earlier, again observe that, C is A t minus 1 and we also know that, C is a subset of B. Notice that, all the indices in B minus A and in A n minus B are greater than or equal to t hence, everything smaller than t, must be in the common region. Hence, our particular set C must be contained in the intersection of the two now, we must have found that, A t minus 1 union x t does not belong to I because we rejected x t from A n.

So, at this stage, we must have found that, this union is not in I but what we notice is that, A t minus 1 is in B, we also notices that, x t is also in B. So, we notice that A t minus 1 union x t is the subset of B and from the first condition of matriods, from condition 1, A t minus 1 union x t must belong to I. Once again what we have found is, there is a contradiction hence, our assumption must be wrong. And since s and t are distinct, the only option left is that s is less than t, strictly less than t so that is the objective of the second claim, which we have established.

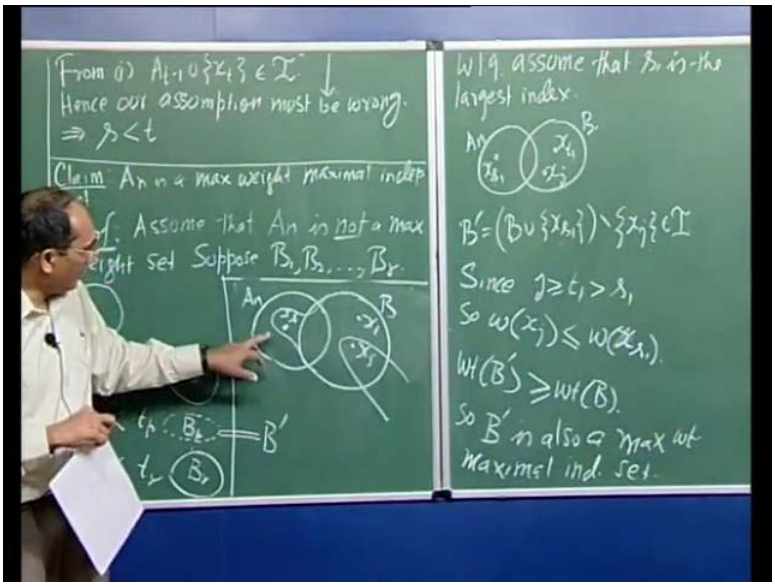
(Refer Slide Time: 28:09)



Now finally, we will prove claim will establish, let A n is indeed a maximum weight maximal independent set. Now, to prove this, let us look at those sets which are maximal maximum weight among the maximal independent sets. If A n is already in this then there is nothing to prove so let us assume that, A n is not a max weight set so there has to be atleast one maximum weight maximal independent set in the collection. So, suppose, B 1, B 2, B r these are all maximal independent sets with maximum weight notice that, r has to be at least one.

So, let us just look at the picture here, we have A n and then against each B 1 through B r, there is some s 1 and some t 1 upto s r and t r. This s 1 is in A n minus B 1, it is the smallest index among them, t 1 is the smallest index in B 1 minus A n and we have already shown that, this is the inequality. So, we will claim, that the same is true with each one of these sets now, among all these, let me pick the largest s value.

(Refer Slide Time: 31:28)

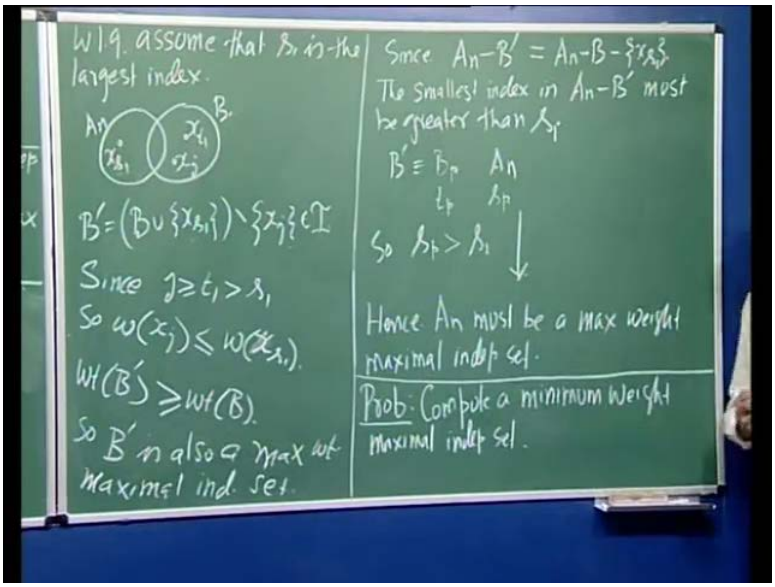


So, among these s 1 through s r without loss of generality assume that, s 1 is largest, without loss of generality suppose, we assume that, s 1 is the largest index. Now, look at this set B 1, we have x s 1 and we have x t 1 so from condition 2 of the matriods, B prime which is B union x s 1 minus some element x j, also belongs to I where, x j is some element in this region. So, it could be x t 1 or it could be any other element in this part now, since j is greater than equal to t 1 because by choice, t 1 was the smallest index in this region.

And we have already proved that, x is less than t 1, s 1 is less than t 1 so we have j greater than s 1. So, what we notice is, that the weight of this element is greater than equal to weight of this element because our indexing is in non increasing fashion. So, the weight of x j is less than or equal to weight of x s 1 hence, the weight of B prime is at least as much as the weight of B. Now, B is one of the maximum weight maximal independent set so we conclude that, B prime is also a max weight maximal independent set.

So, let us come back to this picture and what we conclude is, that the set B prime must be one of these so the question is, what would be the corresponding s i and t i. So, let me just plug in one set here, call it B p, we have s p here, we have t p here and of course, that inequality holds. And let this be same as our B prime, that we have calculated now, in our picture, x s, x t, x j what we have done is, we have deleted this and we have included x s in our B prime. Now, if you look at B prime minus A n then the smallest index in B prime minus A n must be strictly larger than s.

(Refer Slide Time: 36:38)



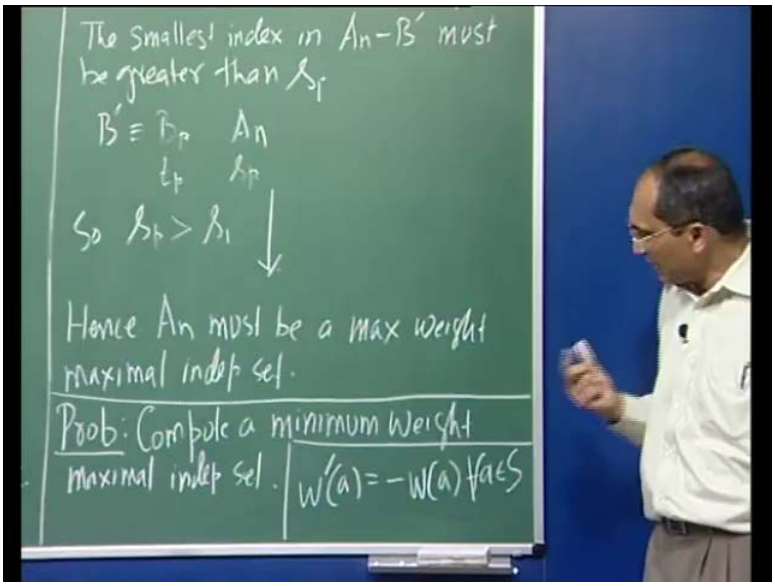
So, I will write down that, since A n minus B prime is actually A n minus B minus x s hence, the smallest index in A n minus B prime must be greater than s. Now, we had labeled our B prime as B p, our B prime a same as B p and the corresponding smallest indices in their complementary regions where, t p and s p. So, from this observation, we

conclude that, s p is strictly larger than s, we had been saying s 1 so I will call it s 1, this is larger than this.

Now, if you recall the choice of s 1 was based on the fact that, this was the largest index in the list of a size that we had listed that but we now have a bigger s then s 1. So, this is again a contradiction, this is observed and hence, our assumption that, A n is not a max weight maximal independent set must be wrong. Hence, A n must be a max weight maximal independent set so we have now establish, that the algorithm directly computes the desired object namely, one of the maximum weight maximal independent set.

So now, here is one question for you suppose, I want to compute, the problem now is, I want to compute one of the minimum weight maximal independent set, compute a minimum weight maximal independent set. So, the only change here is that, instead of maximum I want to compute the minimum weight so how do we solve this problem. Now, the answer is very simple, observe that we are free to choose negative and positive weights hence, we can actually invert the weights, if we replace the weights by their negation.

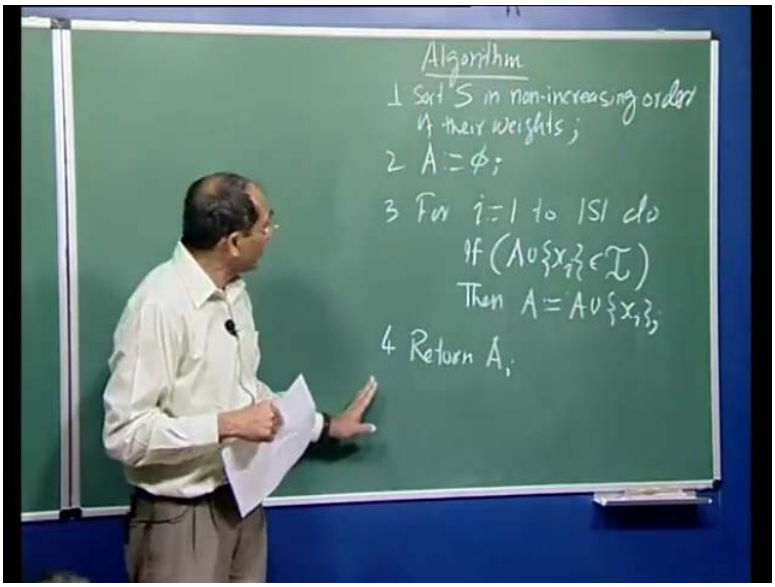
(Refer Slide Time: 46:54)



So suppose, I define a new matriod where, the weight function is different, everything else is same, at this of a is equal to minus of W of a for every a in S. We just do that and then run over algorithm to compute a maximum weight in new sense, we will get a minimum weight in the old. The implication of this observation is, that in our algorithm

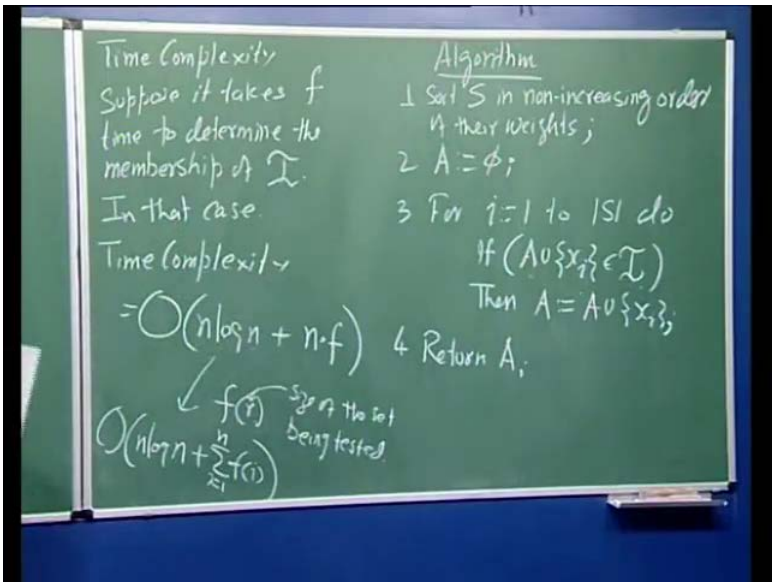
the first step where, we are sorting the elements of S, for minimum weight we will have to sort in the reverse order, we will have to sort in non decreasing fashion. So, the first element will have minimum weight, next will have the next and so on hence, we can solve both the maximum and minimum problem.

(Refer Slide Time: 42:04)



So now, just look at the time complexity of this ((Refer Time: 42:05)) algorithm notice that, we do not have the answer to the question of membership. We do not know, how much would it cost, it depends on how our matriod is represented.

(Refer Slide Time: 42:24)

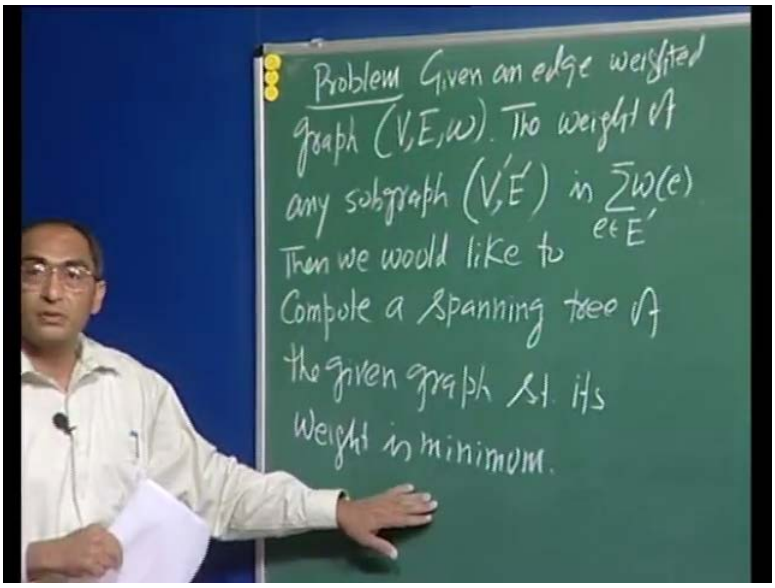


So, let us suppose, it takes f time to determine the membership of I, if I assume f is independent of the size of the set, for which we are checking the membership. In that case, the time complexity is n log n, this is because we have to do sorting, n is the total size or cardinality of S plus in each step, we do one membership check and one constant operation. So, may be we can say, this is the total cost of the for loop, so the cost looks like this.

In case, f depended on the size of the set we are testing, that could happen then let us say, the cost is f r where, r is the size of the set being tested. Then this cost would look like order n log n plus sum f i, i going from 1 to n notice that, in i th iteration the set we are testing, cannot have size greater than i. So, this would be an upper bound to the total cost of computing the set A n or the final set A. Now, what we will do is, we will discuss one application of this abstract structure in graph theory and show that, this simple algorithm computes a desired object in that case.

As one comment, that I want to make in this about this algorithm is that, this is a greedy algorithm, every time we make a decision about whether x i must be in the final set or not. We do one test and then finally, put this back into the set A and we never go back on it so the decision is made then and there, we do not wait for some other computation to make this decision, whether x i should go into the set A or not. So now, I am going to just introduce a problem in graph theory and then in the next lecture, we will show how this algorithm works for that graph.

(Refer Slide Time: 46:33)



The problem that we will discuss in the next class is, computing a spanning tree in a edge weighted graph such that, the weight of the spanning tree is minimum. Suppose, we are given an edge weighted graph V E w, the weight of any sub graph V prime comma E prime is sum w e for e belonging to E prime. So, we define the weight of any sub graph as the sum of the weights of the edges of that sub graph. Then, we would like to compute a spanning tree of the given graph with a condition, that the weight of that spanning tree is minimum. Such that, it is weight is minimum, this is the problem that we would like to address in the next lecture.