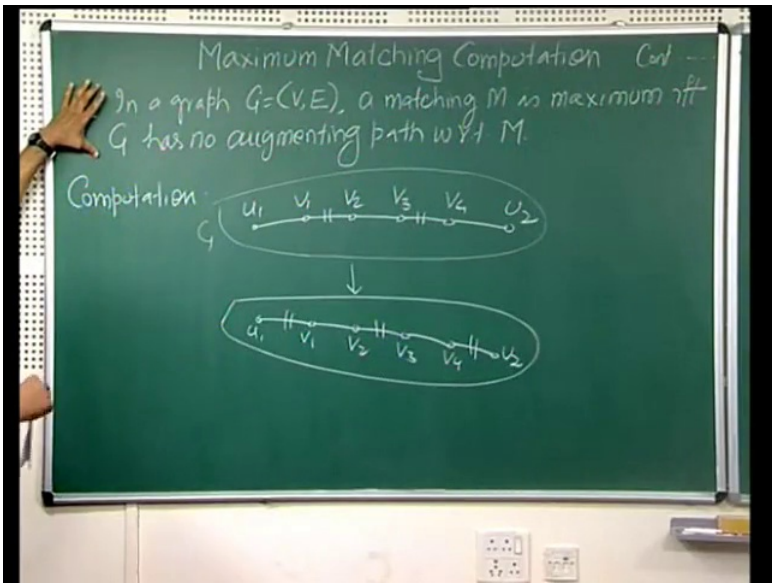
Computer Algorithm - II
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 08
Edmond's Matching Algorithm –II

Hello, so we will continue our discussion of computing a maximum matching in a graph in the previous lecture we had a few results.

(Refer Slide Time: 00:29)



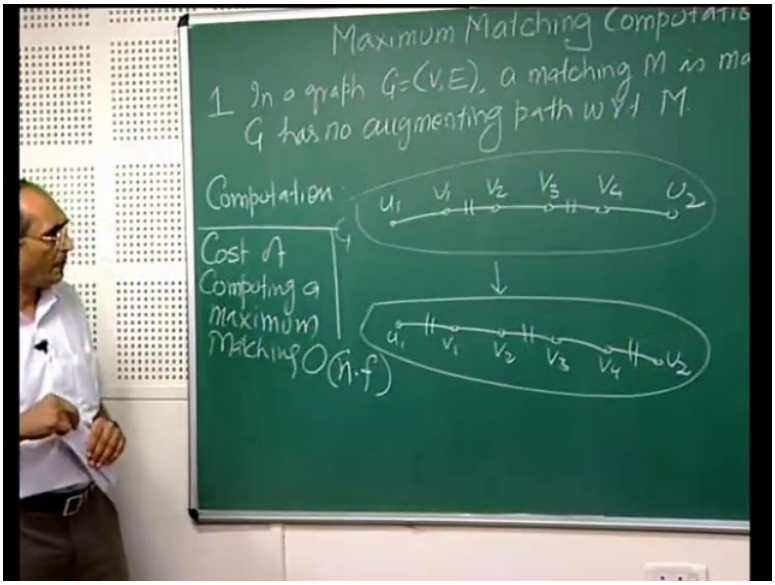
The first 1 that we described was a characterize of a maximum matching, it says that in a graph G V E a matching m is maximum if and only if G has no augmenting path, with respect to m. So, this allows you to know whether a matching have is a maximum one not provided you can compute an augmenting path or decide that there is no augmenting path in the graph. So, the problem reduces to compute in augmenting path, but the question is how does that help you compute a maximum matching. So, for the computation, if you have an augmenting path let say in the graph where v 1 v 2 and v 3 v 4 are the matching edges in m.

So, this is in the side the graph G then we can modify this into another matching with u 1 v 1 v 2 v 3 and v 4 v 2 has the matching edges and those original matching edges have been rendered non matching edges. One can verify that this set of matching edges along with the other matching edges in the graph, still constitutes a valid matching. But the

advantage is that we have augmented that matching, the new matching has one more edge. This way we can improve the size of the matching every time we can compute an augmenting path. So, this is not only a characterization, but also a way to compute a maximum matching.

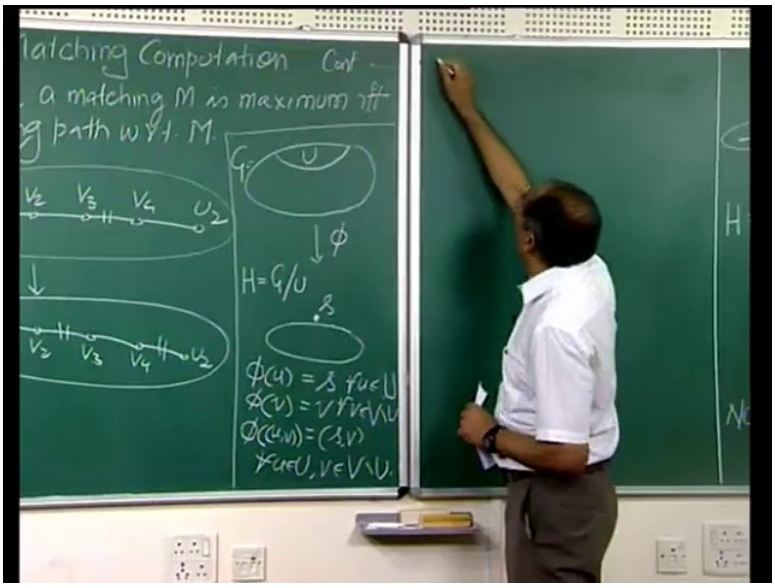
Now, so the problem now reduces to compute an augmenting path, now in each step when we compute an augmenting path, we can improve the matching size by one and the matching cannot have more than n by 2 edges, and bring the number of vertices. So, one may have to compute this n by 2 times.

(Refer Slide Time: 04:10)



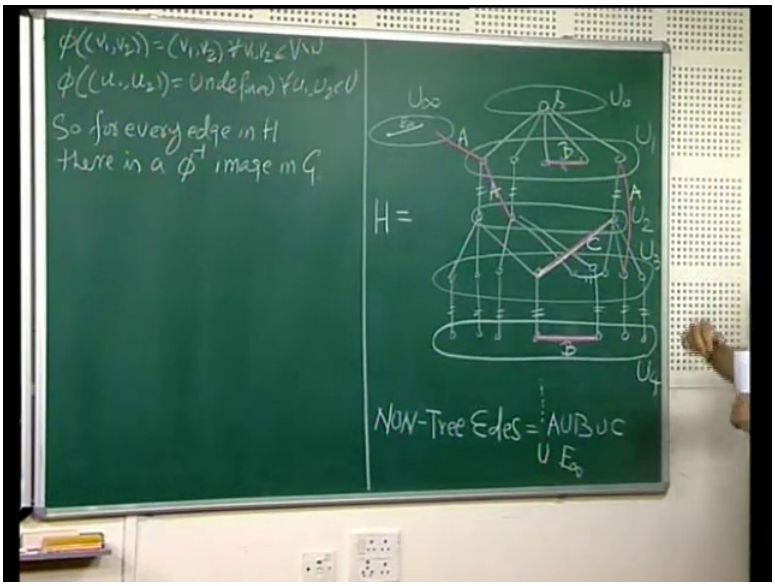
So, the cost of the computing a maximum matching will be order n times f, where f is the cost of computing a an augmenting path. The second thing we did we described a graph based on the original graph and the given matching M.

(Refer Slide Time: 04:57)



So, suppose we have graph G and these symbolically denote the set of un matched vertices the u is the set of matched vertices and this is graph G. Then we defined graph H which we denoted by G modulo U has this where we contract all the vertices of u into a single vertex s. And do not modify these vertices and edges so for symbolic simplicity let me just denote that this is a map, how does that map the vertices. So, V sorry any u in capital U maps to s any V match to V for all V in V minus U we have made a mistake, this is capital U. Any un matched vertex goes to s any matched vertex goes to itself in addition psi of U V is s V for all U, U and V in V minus U.

(Refer Slide Time: 06:48)

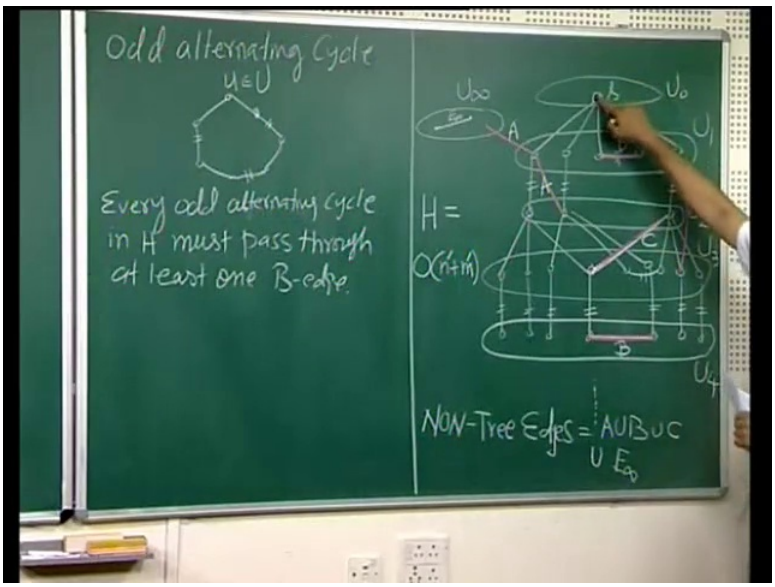


Further psi of v, v, v 1 v 2 goes to v 1 v 2 for all v 1 and v 2 in V minus U and finally, five of u 1 u 2 is un defined for all u 1 u 2 in u. This is effectively what we achieve here, so note that as for as the edges of this graph or consult, which are not contained in the set. So, the edges of these 2 type the u v type and v 1 v 2 type they have a unique image here. And therefore, we can map backwards so for every edges in H this is a inverse image in G.

So, this something we will need to use make a round that defines our new graph, which is sort of reduces which has got pure vertices in general. Of course, there is even vertex here then that graph is the same as H well then taking this graph H, we define a alternating path tree on it. So, lets us take a look at that which we had describe earlier, the set u not contains a single vertex s all its reverse are in U 1, there matching the vertices are in set U 2. All the new neighbours are in U 3 and so on. So, this has got non matching, matching non matching, matching and so on edges. As these are the three edges in addition to that whatever the edges in H we have we partition them as follows.

Those edges which are between 2 vertices of set U infinity they are labeled as E infinity here, those edges which run from an odd level either into U infinity or any level below this U 1. So, it could go to U 2 or U 3, U 4 anything down, they are labeled A and any edge non tree edge, which runs from an even level to an odd level are labeled c in addition to that any edge which runs within the vertices of given level.

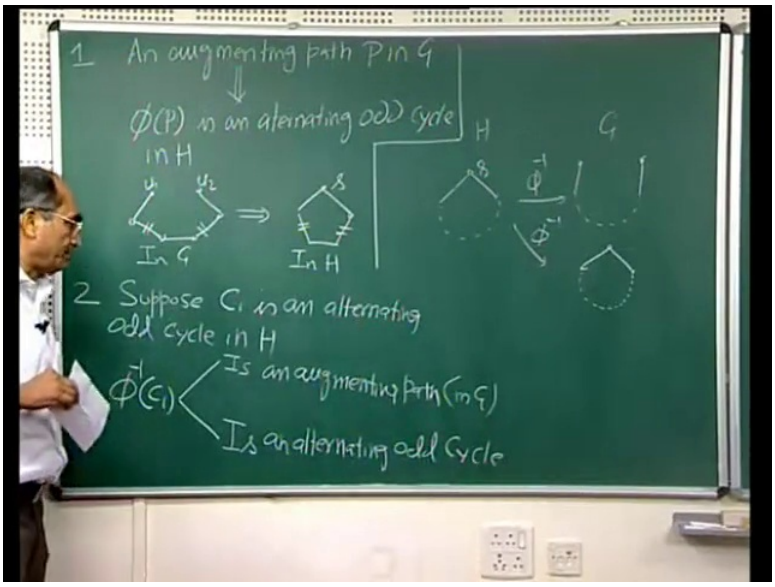
(Refer Slide Time: 11:45)



Let say in case of odd levels it will be a matching edge in case of and even level this will be a non matching edge these are all labeled B. So, the non tree edges are partition into these four classes we also showed that this tree can be computed using breadth for search. And we can also identify all the B edges during this computation so this entire computation of computing.

This tree as well as the B edges takes order n plus n time where n is the number of vertices m is the number of edges in H. So, maybe I will use prime to we would differentiate between those numbers for G, so this much we had done and now we are prepared to proceed with the actual computation of an augmenting path. We had define something called odd alternating cycle, which meant in odd cycle where there is a vertex U which belongs to the set U, which is un matched following which we have alternatively matching and non matching edges in it. So, these 2 edges are obviously non matching edges and then we have the alternating path from this vertex to this vertex. And we had shown here that every odd alternating cycle in H must pass through at least 1 B edge.

(Refer Slide Time: 14:57)



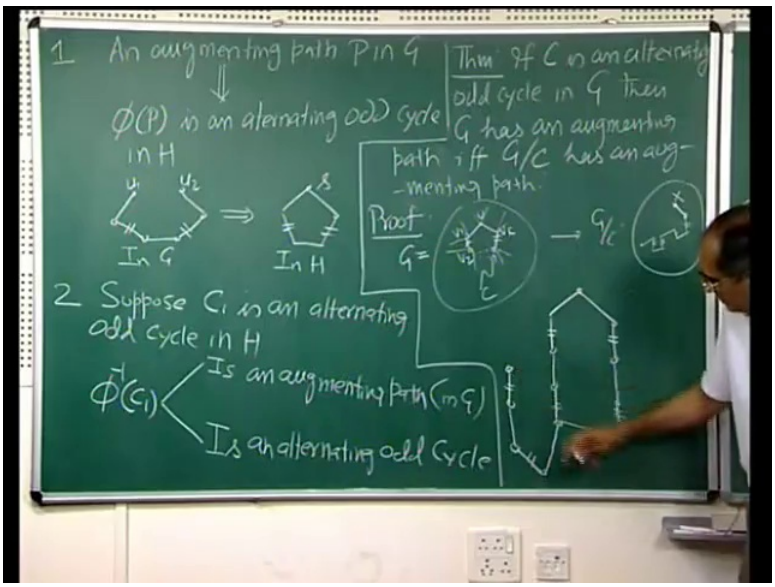
Now please note the every odd alternative cycle in H must pass through s because this is the only un matched vertex in this graph, all the matching edges of G are the matching edges of H. So, we have really a graph h m within a views of symbol same m that we have used in G, so an alternating odd cycle will pass through s and the claim says that it must pass through at least 1 B H. So, we are have this information and now we will

begin the discussion from for the computation of an augmenting path. So, let us first observe that an augmenting path in G.

An augmenting path call it P becomes alternating odd cycle in H. So, you take an augmenting path in P in G, what you notice is that its image. For example, if this was the path this is U 1 and U 2 these are the un matched vertices then this will transform into an alternating an odd cycle in H this is in G, but what happens if we have an alternating odd cycle in H can we recover a augmenting path in G from that. Well we have something along those lines, which we will use in order to compute an augmenting path in G. So, let us suppose C 1 is an alternating odd cycle in H.

Then what is its image in G remember that cycle is a set of edges so an edges image have of psi inverse image in G. So, this is a well defined entity, this passes through s, so there are 2 possibilities clearly that this is obviously in G. So, if I do the inverse in H of this it is possible that let we do psi inverse of on this these two images may end up in two different matched vertices. So, that with third into this path and this is an augmenting path, in other situation this is not the case and this is still an alternating odd cycle.

(Refer Slide Time: 20:00)



In case both of these edges were incident on the same vertex in G, then this remains as it is of course, we go to that corresponding vertex so in that case we have still this situation in G this in H and this is in G and we have this picture. So, what do we do in case we

discover such a cycle, how does that help us figure out an augmenting path in the graph G, so then there is a result that we will show.

That if C is an alternating odd cycle in G then G has an augmenting path if and only if G mod C has an augmenting path. So, what we notice is that either suppose we compute a augmenting odd cycle in H then when we do psi inverse either we directly get an augmenting path. And if we do not and we end up getting an augmenting alternating odd cycle in G then we will explored this result, what it is says is that there is an augmenting path in G if and only if there is an augmenting path in G mod C. Which means vertices of C have been strong into a single vertex.

So, this is a smaller graph we will recursively call the algorithm for computing an augmenting path in this graph. And then will show that not only that this gives an augmenting path, but you can actually build an augmenting path for G using this path. So, we will actually recover such an augmenting path in G, so let us try in prove this claim. So, let suppose we have a graph G there is a path cycle C which is odd and it is an alternative cycle this is the story in G.

So, G mod C this is cycle icon C so mod C is some vertex x this is the vertex that results from shrinking these vertices of the cycle into single vertex, we do not touch anything else that se as it is. Now, first let see there is an augmenting path here, let suppose there is an augmenting path which in this graph, which does not touch x it does not touch x. So, this vary path is also present here and it is also an augmenting path so we immediately see that there is an augmenting path of this type then there is 1 in G.

Now, alternatively there is an occurrence of x on this path so x occurs in on this path. Note that all the vertices are other than U or matched vertices see then there matching edges are on the cycle. So, none of these edges that emerge from these vertices are matching. So, what you notice is that the vertex x in this graph is an un matched vertex.

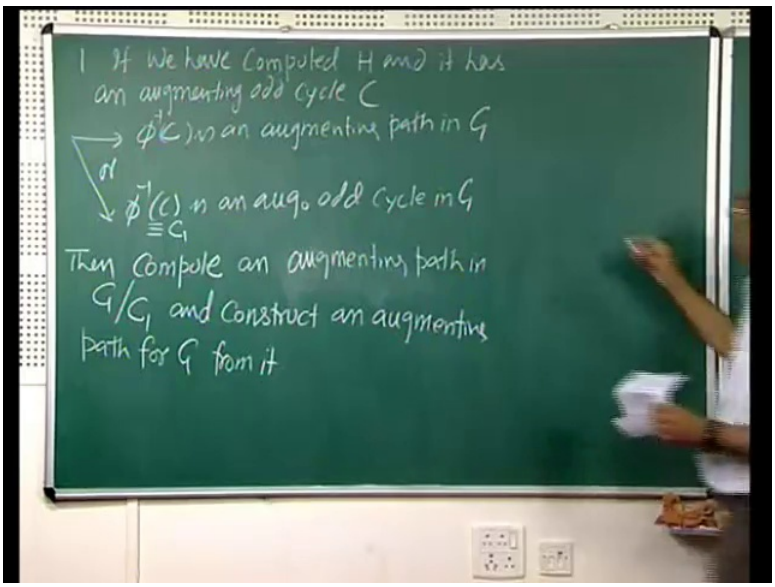
Un matched vertex can occur on an augmenting path only at 1 at 1 of the 2 ends, so that indicates that if x is on this path that it has to be like this. If the path is this where one of the end vertices is x then that is compute x inverse. So, inverse may look like the following just for this take of an example I am drawing one, this is the cycle C and we have a path, when we compute the inverse the lost edge that is connected to x will

actually go an end of in one of the vertices of the cycle. So, let us assume it is one of these vertices say this vertex in particular.

So, you have now this picture what do we do with it, how do you recover a an augmenting path. Now, as such this will not give you an augmenting path in the original matching. So, original matching m does not provide an augmenting path out of this structure. Well actually we does my mistake you can directly take this up to this, so there is a augmenting path write here to take I should start from an un matched vertex here.

So, starting from an un matched vertex this path goes up to some vertex here and then there is one of the two vertices will be a matching edge. So, you follow the alternative if this is actually an un matched edge this is the un matched edge that followed by the matching and un matching. And you continue until you find an U, so if this is V we have found a an alternating path with from u 2 and V 2 u and these vertices un match so it is an augmenting path. So we really do not have to do anything else it does compute an augmenting path directly from this.

(Refer Slide Time: 27:37)

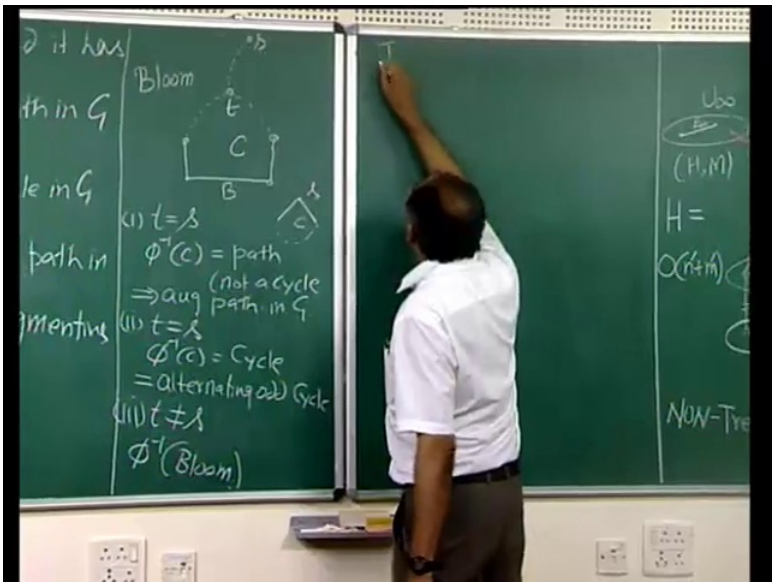


So, let me quickly summaries what we have found is one that if we have computed H and it has an augmenting odd cycle C, then either psi inverse C is an augmenting path in G or psi inverse C is augmenting odd cycle in G. Then we call this C 1 then the problem reduces to the following then compute an alternating an augmenting path rather

competence augmenting path in G modulo C 1. And construct an augmenting path for G from it this is the plan.

So, the question now is how to compute an augmenting odd cycle in H. Now, let us go back and recall that every alternating odd cycle in the graph H must pass through B H. So, let us take a look at such a B H what you do is you take it is two end vertices, and start walking along the 3 edges upwards from both. In this case what happens is derive these two edges they converging immediately into s. So, you find an odd cycle which is an alternating odd cycle in this case again we walk along the 3 edges. Well fortunately again they meet at s and again we have found a an alternative odd cycle, but in general they might meet at a vertex at a lower level in the 3.

(Refer Slide Time: 31:43)



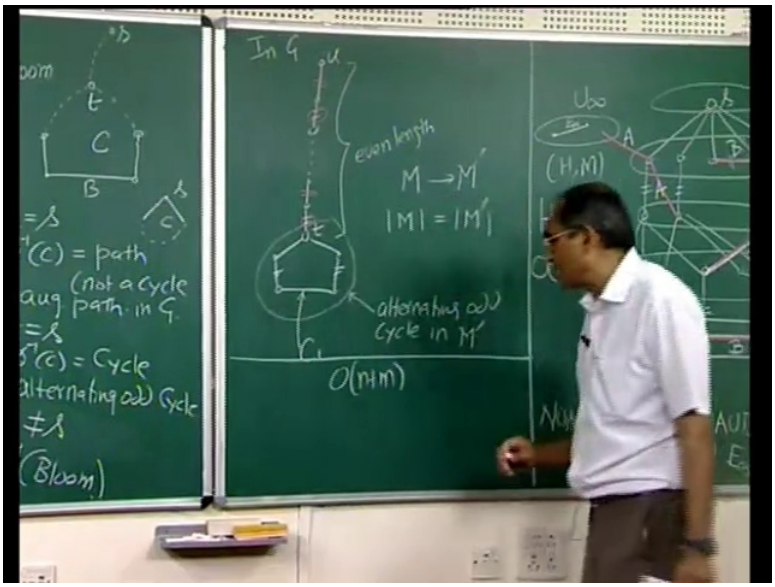
So, let us take a look at what happens when we start from a B H and starts raising. And ultimately we meet at a vertex call it t and from there, there is a unique tree path which takes you to s. Let’s focus on this structure this is called a bloom an upside down this is the stem and this is the flower. So, we can compute this in order n time because we already have computed the alternating path 3 from H just tracing the 3 edges, we will determine the bloom.

Our goal is to compute an alternating odd cycle or alternating, or augmenting path in the graph G and for that it would be find if we succeeding in computing an alternating odd cycle in H. So, there are three possibility first one the t is equal to s like we saw in our

examples both those examples had the situation that t was same as s. Now, let us take look at so I will use the symbol C to indicate the cycle in this, so what if psi inverse c if you look at the psi inverse c. And remember this is s hence this is the un matched vertex the only matched vertex in H.

So, this is clearly an alternating odd cycle in this is a path not a cycle which means the pre images of here is the situation this is s and we have c here the 2 edges are incident on 2 distinct vertices of set u in G, then this is a path then this is an alternating path or augmenting path. So, we directly find what we wanted in G, but second situation is the t is equal to s, but psi inverse c is a cycle which means, even when we map them back into g these 2 edges are incident on the same vertex. So, this is a alternating odd cycle, we know how to proceed from this point, we already have seen that if we this is an alternative odd cycle. Call it C 1 then all we have to reduce the problem to computing a augmenting path in graph g mod C 1. The third option then is that t is not equal to s in this case when we take psi inverse of the bloom.

(Refer Slide Time: 35:55)



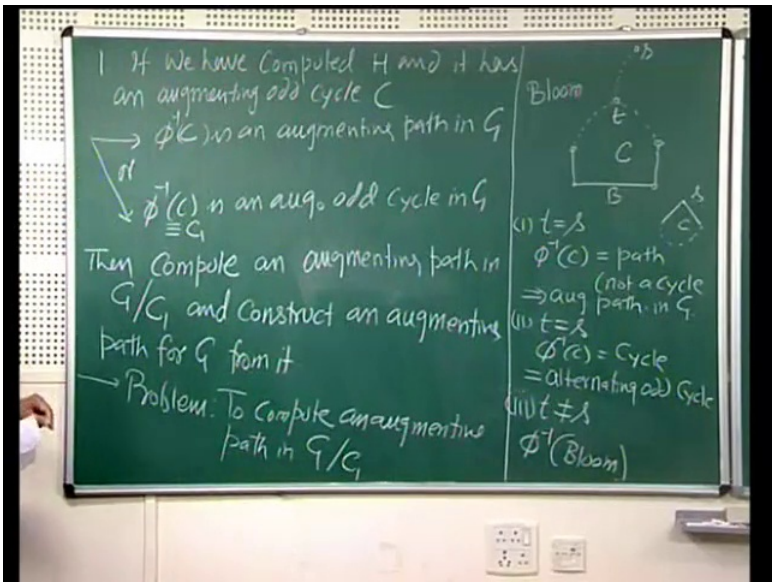
What we find in G is some vertex U will be lets us take as an example his is the situation, but this is we vertex t which is one of the matched vertices, s is replaced by some un matched vertex in G. The stem as even length this is an odd cycle, where from this point to this point is an alternating path. And these two are matched edges now in this case when we found this structure in G, we will do the following we will modify our original

matching m into another matching m prime in the following fashion, we will remove this and make this a matching edge.

So, we will take the alternating edges which were un matched make the matching edges and the matching edges on the path happen rendered matching edges. So, we find a alternative matching this is still matching a matching, where all the other edges are has it is. Note that this is an even length path so the number of edges which have been made matching is equal to the number of edges, which were non matching may be turn into non matching.

So, the size of m is equal to the size of m prime, what is interesting about this is that this U has been now turn into a matched vertex and t has been turn into an matched vertex. In this matching M prime this is now is an alternating odd cycle in M prime. So, what we have found is in the first is in the first option directly get an augmenting path, but in second and third option we have an alternating odd cycle in G with respect to some matching may be the original matching M or a new matching which has the same size. Now, with this call this is C 1 we have the following problem.

(Refer Slide Time: 39:13)

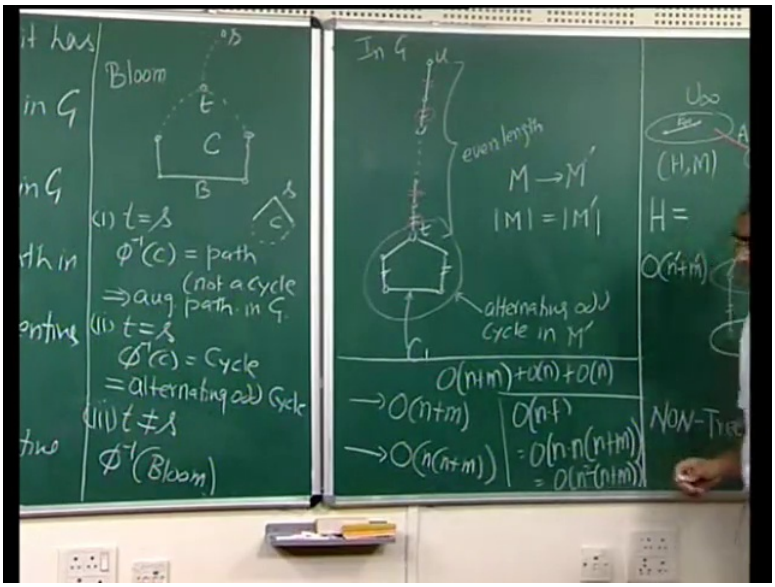


Now, the problem is to compute augmenting path in G modulus C 1. So, we have reduce the original problem of computing an augmenting path in G into this problem is an odd cycle has at least 3 edges. So, 3 vertices then we into a single vertex at least few vertices are fewer in this graph then in the origin graph G. So, this is a smaller graph with at most

n minus 2 vertices in it. Hence we have reduce the original problem into a smaller problem other we have already solved it or reduce this into this problem.

So, this step this recursive procedure we go on for at most n by 2 recursive calls, so let us take look at the total cost. Computing the alternating path tree from H takes order n plus m time so n prime and m prime may be less than equal to original n and m so let us not worry about it will just take the worst case. We take order n plus m time to compute the tree from the tree we construct the bloom in order n time and from the bloom we go back compute the inverse is. And if necessary modify the matching's and compute an alternating odd cycle with respect to original or a modified matching.

(Refer Slide Time: 41:38)



So, all this process would we additional order n time because the number of edges involved here or order n. So, this entire first step is order n plus m by this we have already computed the augmenting path or we are going to now recall this procedure. And this will go on for order n time and after the maximum number of such recalls of the procedure, we have taken this much time at most to go down to the lowest level we have either discovered an augmenting path or we have concluded that there is no augmenting path. So, one step in the whole procedure of computing an augmenting path takes this much time. Earlier we had already shown that the total cost is order n times f, which is order n times n times n plus m, which is order n square n plus m.