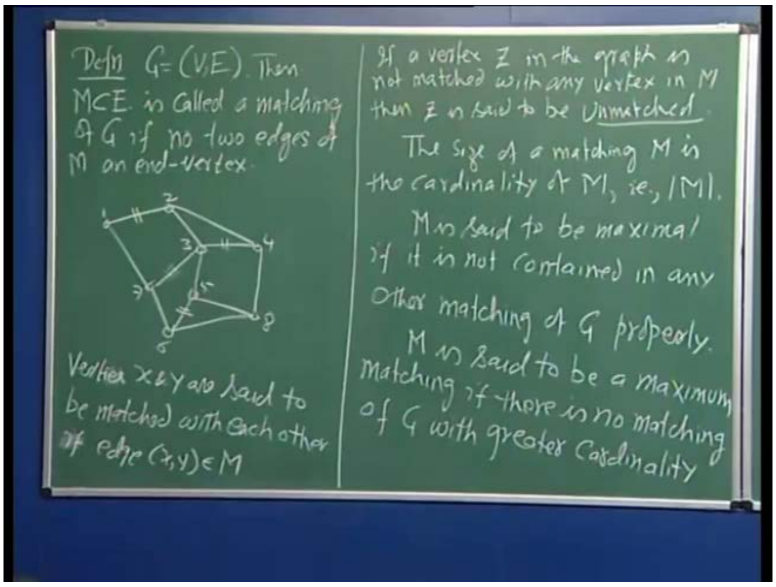
# Computer Algorithms - 2

**Prof. Dr. Shashank K. Mehta**
**Department of Computer Science and Engineering**
**Indian institute of Technology, Kanpur**

## Lecture - 7
## Edmond's Matching Algo I

(Refer Slide Time: 00:27)



Defn $G=(V,E)$. Then $M \subseteq E$ is called a matching of G if no two edges of M an end-vertex.

Vertices x & y are said to be matched with each other if edge $(x,y) \in M$

If a vertex Z in the graph is not matched with any vertex in M then Z is said to be Unmatched.

The size of a matching M is the cardinality of M, ie, $|M|$.

M is said to be maximal if it is not contained in any other matching of G properly.

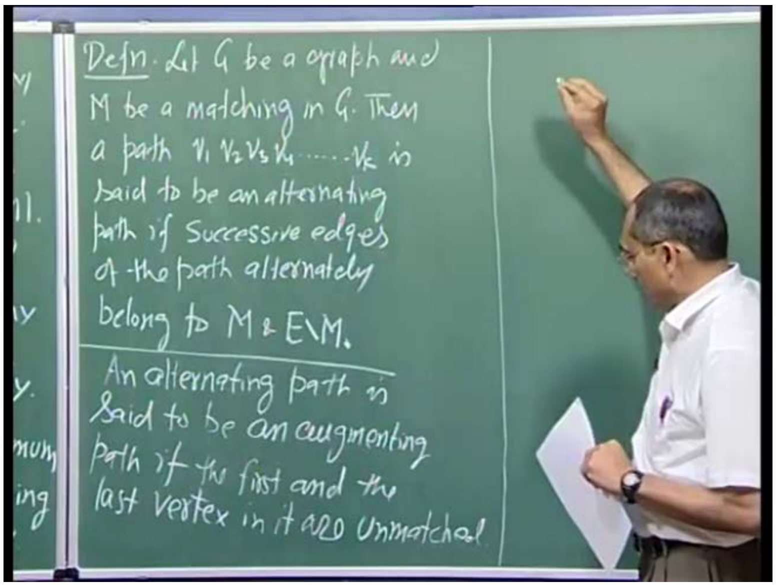
M is said to be a maximum matching if there is no matching of G with greater cardinality

Hello. Today, we will discuss in algorithm to compute on matching in a graph. So, let us begin some definition. Let us suppose, G is an undirected graph, then if sub set M of the edges is called a matching of G, if no 2 edges of M share the same end vertex. So, let us take an example. Suppose, we have a graph like this, in that case these 3 edges which are marked by these lines constitute a set M that is a matching. The reason is the n vertices of these edges, namely these 2. Let us label them 1 and 2, 3 and 4, 5 and 6. None of the n vertices of the 3 edges are common.

This is called matching, because you are matching 2 vertices or paring of 2 vertices. In this case we have got such 3 pairs. Then we say that vertices X and Y are said to be matched with each other, if edge X, Y belongs to the matching. If a vertex Z in the graph is not matched with any vertex in M, then Z is said to be unmatched. The size of a matching M is the cardinality of M. So, the number of edges in the set M is called the size of the matching. Now, we are also interested in the size. So, if we say nothing is maximal, that could mean that M is in not properly contained in any other matching. So, M is said to be maximal, if it is not contained in any other matching of G properly.

Besides, a matching is said to be maximum if there is no matching with greater cardinality. M is said to be a maximum matching, if there is no matching of G with greater size. Then it is obvious that every maximum matching is bound to be a maximal matching. So, in this case for example, if I add 1 more edge this is a maximal matching because I cannot add any more edge to M without violating the condition of sharing in end vertex. This not valid in fact, this is a maximum value. You cannot add any more, but this need not be a maximum matching, that we will lecture in later on and try to figure out another important definition, which is that of an alternating path.

(Refer Slide Time: 07:49)



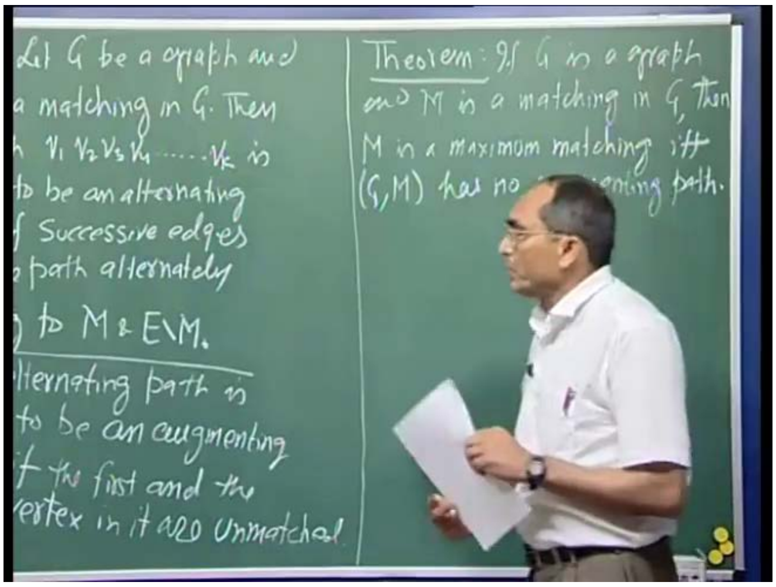
Defn. Let G be a graph and M be a matching in G. Then a path $v_1 v_2 v_3 v_4 \ldots\ldots v_k$ is said to be an alternating path if successive edges of the path alternately belong to M & $E \setminus M$.

An alternating path is said to be an augmenting path if the first and the last vertex in it are unmatched

Let G be a graph and M be a matching in it. Then we can define alternating path in the following fashion. Then a path V 1, V 2, V 3, V 4, V k is said to be an alternating path, if the first edge is a non matching edge, second is a matching edge, third is a non matching edge, fourth is a matching edge and so on or the first is a matching edge, second is a non matching edge. Then again a matching edge and so on, if successive edges of the path alternately belong to M and E minus M.

Note, in this contest we define 1 more notion and the notion of an augmenting path. An alternating path is said to be an augmenting path if the first and the last vertex in it are non matching. In that case, if this and this are non matching vertices, then clearly the first edge has to be a non matching edge then of course, a matching and a non matching and it should end with a non matching edge. Now, we will describe a useful result which will help us to compute a maximum matching.

(Refer Slide Time: 11:28)



Let G be a graph and
a matching in G. Then
h $v_1 v_2 v_3 v_4 \ldots v_k$ is
to be an alternating
f successive edges
path alternately
to M & E\M.

lternating path is
to be an augmenting
f the first and the
ertex in it are unmatched

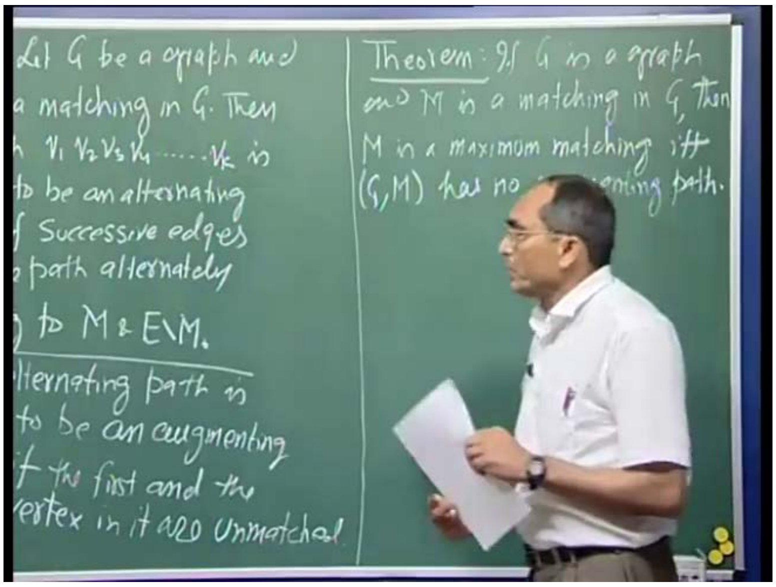
Theorem: If G is a graph and M is a matching in G, Then M is a maximum matching iff (G,M) has no enting path.

The result is that, if G is a graph and M is a matching in G, then M is a maximum matching if and only if G coma M has no augmenting path. So, let us go back to the figure. In this figure, there are 2, 1 and 2 unmatched vertices. So, if you look at this, if you walk along 7 3 4 8, what you find is an alternating path, where the first and the last are unmatched. So, according to theorem this is not a maximum matching. All the way not going to prove that theorem, I can show you 1 important connection as to how I can enhance this matching. The moment I found an augmenting path, I will switch the edges. The non matching edges will be put in the matching set and the matching edges in the non matching.

So, this is how I am going to switch the edges and what you notice is the resulting matching is still a valid matching. But, we have 1 more matching edge. There are 4 matching edges. As this is a bigger matching than the previous one, now in this case it is very obvious that this is a maximum matching because there are only 8 vertices and 4 matching edges. Every vertex has been matched now. Now, had it been a bigger graph may be this was not a maximum matching. In that case, we would have to look for another augmenting path.

If we continue to switch the edges on the augmenting path, we will eventually end up with the maximum matching because in each step we are going to improve the sizes of the matching. Since, now we have found a way to compute a maximum matching, but before that we need to prepare some more background. So, here we need to define some thing called an odd alternating cycle.

(Refer Slide Time: 15:02)



Let G be a graph and
a matching in G. Then
h $v_1 v_2 v_3 v_4 \ldots v_k$ is
to be an alternating
f successive edges
path alternately
to M & E\M.
lternating path is
to be an augmenting
f the first and the
ertex in it are unmatched

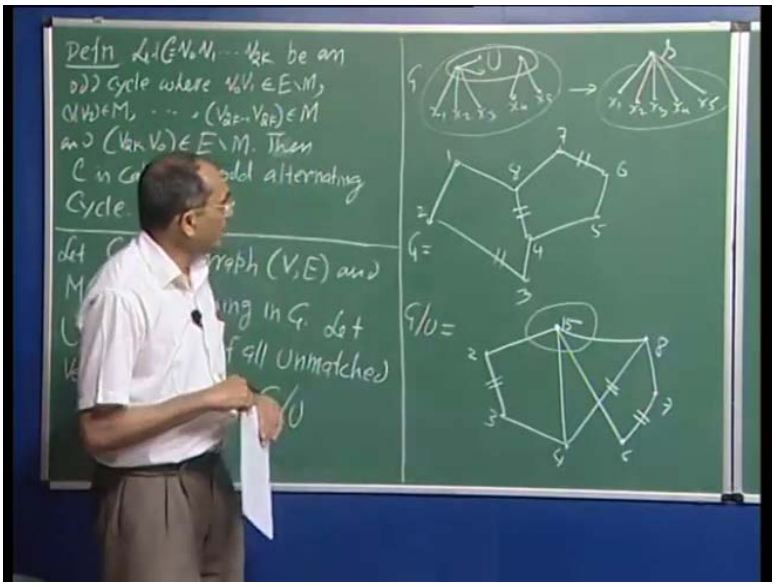
Theorem: If G is a graph
and M is a matching in G, then
M is a maximum matching iff
(G,M) has no enting path.

Let V 0, V 1, V 2 k be an odd cycle, where V 0m V 1 is the non matching edge. Then V 1 V 2 belongs to M. We have in alternating membership, in E minus M and M. The second last edge V 2 k minus 1, where V 2 k belongs to M. V 2 k V 0, is a non matching edge. If we have such a cycle, then we call it such a cycle say C, then C is called an odd alternating cycle.

Now, I am going to talk about a transformation in a graph which we will need to do while searching an alternative cycle. So, let G be a graph V coma E and M be a matching in G. Let U be the set of all unmatched vertices. So, there are no matching edges emerging from any of these vertices and every vertex in V minus U is matched. Then we define a graph H denoted by G mod U.

This graph is as follows. So, if suppose G symbolically is this graph and this is the set of vertices in G which are unmatched, then we perform the following transformation. We squeeze this entire set into a single vertex. Let us call that S and the remaining graph is left untouched. What does that mean? So, suppose you have vertices here which has got some exit going out of U and a few edges which are connecting to vertices in U, in that case we will replace all these by a single vertex. These edges will continue, so let us say, these are vertices x 1, x 2 and x 3, may be here there are more vertices x 4 and x 5. Then, this vertex is connected to every one of these vertices. So, we lose the entire structure inside this collection of vertices, but we preserve every thing here as it. So, let us take simple example.

(Refer Slide Time: 20:06)

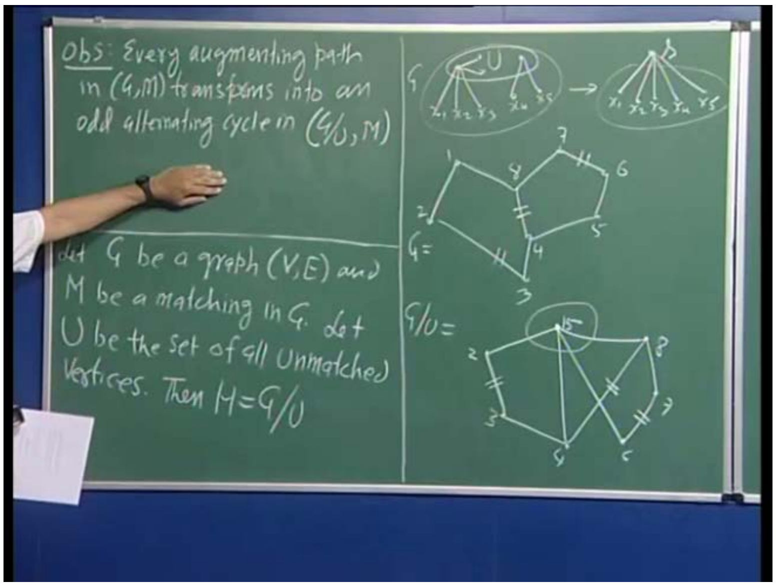


Let us suppose, we have this graph, label them as 1, 2, 3, 4, 5, 6, 7 and 8. So, again coincidentally we have 8 vertices again and 1 and 5 are unmatched. So, if this is G, then G mod U will squeeze vertices 1 and 5. So, let us call this vertex 1, 5 and all the other vertices will stay as it is, vertices 2, 3, 4, 6, 7 and 8. This edge structure among these vertices will be untouched. Now 1, 2 edge will be now the edge between 2 and vertex 1, 5 and 2, 3 will be as it is, that is 3, 4 ; 8, 4 and 4, 5. So we will have this, because there is an edge between 4 and 5. This is an edge between 5 and 6. Now, it will be this one. It is 6, 7. There is an edge between 7 and 8. The edge 6, 7 will be a matching edge, 4 and 8 is the matching edge and 2 and 3 is a matching edge. So, we have now 4 and 5, 5 and 6, 1 and 2, and 1 and 8. So, there is an edge here as well.

This is the result of shrinking the set of unmatched vertices, namely set of 1 and 5. Now, we make a claim. What we want to do is observe what happens to an augmenting path in G when we transform the graph into this. So, let us say this example, suppose we had an augmenting path that would have gone from 1 to 5. In the entire path apart from the first and the last vertex there would not have been any other unmatched vertex. In this graph, of course there are only 2 unmatched, but even otherwise all the intermediate vertices in an augmenting path would have been matched vertices.

So, in this picture that path, for example, an augmenting path in the 1 8 4 5. That path would have started here and ended here itself. So, it could have been 1 4 8 back to this. So, what is that? All the inter vertices are matched and their matching edges are on the path. So, if you leave this alone, then the rest of the path must have even number of vertices because they are all matched and their matching are on the path. So, this has to be an odd cycle and it is the first and the last. It is obviously unmatched and rest of it is an alternating path. Hence, this is an odd alternating cycle. So, we make an observation that every augmenting path in G, M transforms into an odd alternating cycle in G mod U and M.

(Refer Slide Time: 24:49)

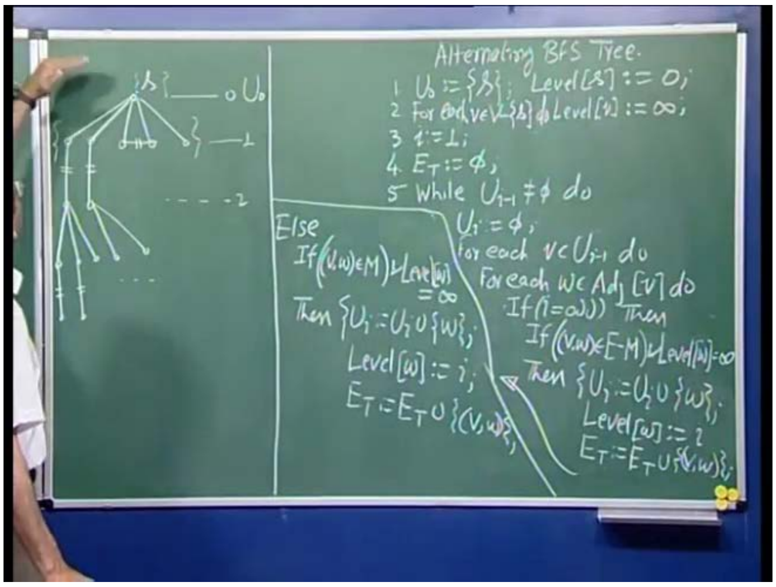


Obs: Every augmenting path in $(G,M)$ transforms into an odd alternating cycle in $(G/U, M)$

Let $G$ be a graph $(V,E)$ and $M$ be a matching in $G$. Let $U$ be the set of all unmatched vertices. Then $H = G/U$

I am using the same symbol M, because after all it is the same set of edges. So, now we would like to first compute some thing called an alternating breakfast tree. An alternating breakfast tree is a structure such as this one.

(Refer Slide Time: 26:12)



Alternating BFS Tree.

```
1  $U_0 := \{s\}$; Level[s] := 0;
2  For each $v \in V - \{s\}$ do Level[v] := $\infty$;
3  $i := 1$;
4  $E_T := \phi$;
5  While $U_{i-1} \neq \phi$ do
       $U_i := \phi$;
       for each $v \in U_{i-1}$ do
           For each $w \in Adj[v]$ do
               If (i = odd) Then
                   If ((v,w) $\in E - M$) & Level[w] = $\infty$
                   Then { $U_i := U_i \cup \{w\}$;
                       Level[w] := i
                       $E_T := E_T \cup \{(v,w)\}$;
               Else
                   If ((v,w) $\in M$) & Level[w] = $\infty$
                   Then { $U_i := U_i \cup \{w\}$;
                       Level[w] := i;
                       $E_T := E_T \cup \{(v,w)\}$;
```

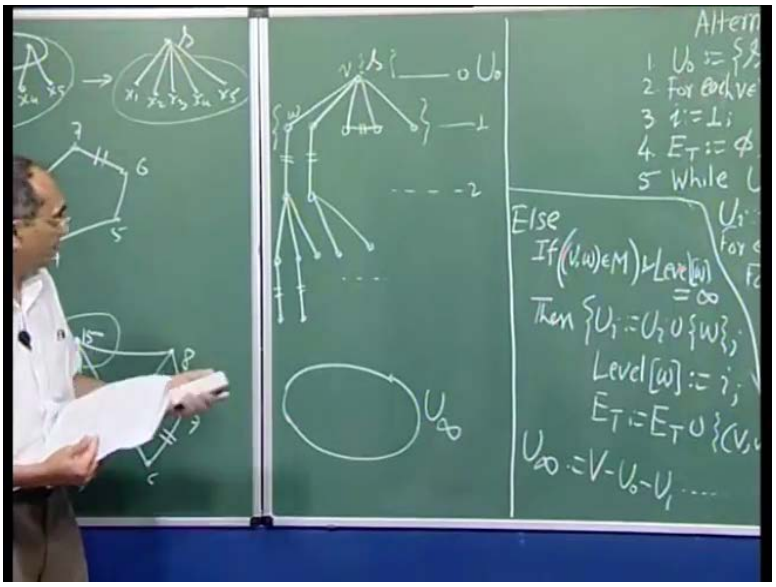
Suppose, we have a vertex S and this vertex is unmatched, then all the vertices adjacent to S are in the first level. I will label them as 0. This is at level 1 and this level could match the edge of this vertex. If there is 1 in the next level, so we will expand and walk alone the matching edge. If the matching edge is here then we cannot expanded further. If there is no matching edge here, then we cannot do anything about it. Then all the new unmatching edges from here are expanded in the next level. Then again if there is a matching edge from here and we will expand to that level and so on. So, we will form such a type where the edges between even and odd level are unmatched edges and from odd to even level are all matching edges.

So, our objective first is to compute such a tree and then we will study the structure of this tree. Now, here is the simple breakfast, say repeat, which has been transformed slightly to be able to compute this structure. What we will do is that, we will use a variable called level for every vertex. We will visualize level of S to be 0 and the set associated with this level will be denoted by U naught. So, U naught contain this. U 1 contains these vertices and so on. So, the first statement is that, set U naught to the single turn and set the level of A and for every other vertex set the level to infinity.

Then, we begin from I equal to 1 onwards. Now, in the process all the 3 edges we will store into a set called E T. So, we initialize it to empty set. Now, as long as previous level is non empty we will continue to compute the next level, so if U i minus 1 is non empty, then I initialize U i to be empty. Now for each vertex in the level, so I take let say, we are looking at this level I take 1 vertex at a time. So, here we pick, say V and U are minus 1. We look at its neighbor and all the vertices adjacent to it. We have to differentiate between the odd and even levels, the reason being we are interested in non matching edges and matching edges here.

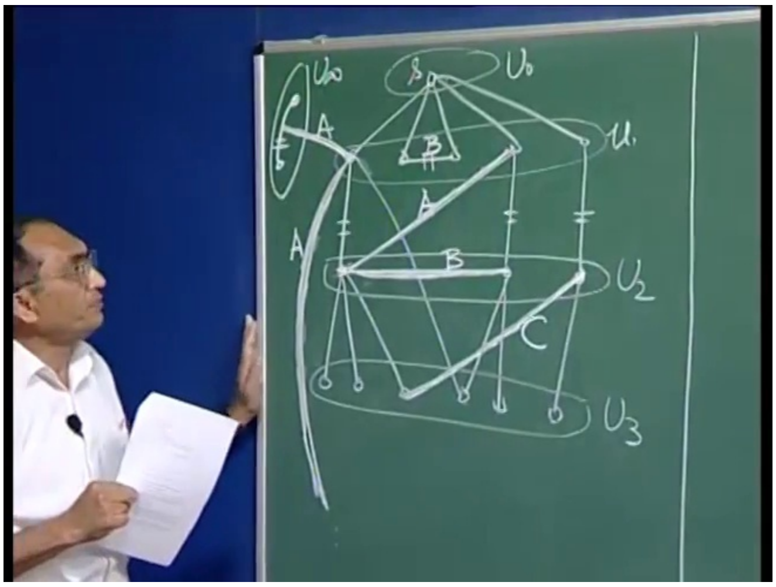
So, we check if i is odd. That means, if you are looking at this level we are trying to go from here to here. Then if V W is a non matching edge, V W is V minus M. The level of W is infinity. That means, if this is V and this is W and level is infinity. It means that it has yet been encountered. In that case, we will incorporate W in U i. We will set its level to i and will tear the edge V W in set E T. Similarly, if the index i is even, that is the X part, then we do the same thing. Only thing is we have to insure that the edge V W is in systematic. The rest is identical. This is how we will be able to partition the vertices into these level. Now, note that there might be some vertices which are reachable by this set.

(Refer Slide Time: 31:44)



Those will remain in a single set called U infinity. So, in this program we should actually say U infinity to be equal to V minus U naught minus U 1. Then we have a partition of the vertices in U naught, U 1, U 2 and ultimately U infinity. Now, let us characterize the edges which are not in the tree. So, let us just go back and build a symbol of a tree.

(Refer Slide Time: 32:25)



If it so happens that the 2 vertices which are adjacent to our starting vertices match together, then obviously the matching edge inside level U 1. This is our U 1, U 0. The next level of course, then have many edges. Suppose, there was a non matching edge emerging from this, then it can provably go like this. It can go to a lower level and then it can go to arbitrarily lower level, to any level. So, may be I can say either this goes any where down. It is also possible that there is a non matching edge which goes to U infinity.

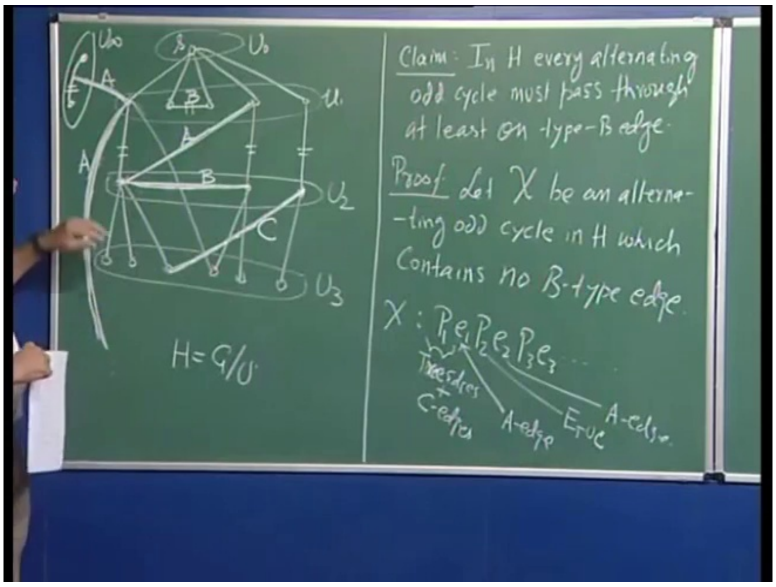
Now, if you look at the even level, then obviously its matching edge is already up there. All the non matching edges are going down, but since this is a tree and this is a breakfast sort of tree, suppose there was a non matching edge connected to this, that will not be part of the tree. So, I will just try to make it like this. Thus initially, we have a set which is also not part of the tree. This is not part of the tree nor is this. Now, it is also possible that there is a non matching edge which runs inside the level.

Then, obviously this edge is not going to be part of the tree. So, what we notice is that there are 3 types of edges. I will categorize them as follows. From an odd level there may be an edge going from the vertex of the odd level under consideration through U infinity. We will consider such edges as type A. In addition, this may be non matching edge because our matching edge is obviously moving down to the next level. There may be a non matching edge going through the next level or to a third level or any where down there. So, these are all characterized as type A.

Then, in an odd level there might be matching edge connecting 2 vertices of the same level. That we will label as a type B edge. Similarly, in even level a non matching edge running across two vertices of the edge. So, from an even level there may be a non matching edge connecting to the next level. Such edge we will label as type C edge. Now, observe that this accounts for all the edge. Now, here you may have to notice that if there is a vertex in this set and say it is a matched vertex, so say it has a matching edge.

Now, that vertex with which it is matched also cannot be part of this graph because had it been here, then that vertex would not have been in even level, because it can only arrive in even level through a matching edge. That vertex could not have been in odd level because if it was an odd level then we would have put this vertex in the next level. So, all the matching edges associated with vertices of U infinity must be only running among them.

(Refer Slide Time: 38:46)



Claim: In H every alternating odd cycle must pass through at least on type-B edge.

Proof: Let X be an alternating odd cycle in H which Contains no B-type edge.

$X : P_1 e_1 P_2 e_2 P_3 e_3 \dots$

Now, we will make a claim. The claim is , let any alternating odd cycle in this graph, please keep in mind this graph has only, we have assumed there is only one unmatched vertex. If there are more unmatched vertices, then the structure must be of different type. But, we are interested in this graph H, which is G mod U and this classification is done for H. Then any alternating odd cycle must start from S and go back to S because this is the only unmatched vertex.

So, the claim is that in H every alternating odd cycle must pass through at least 1 type B edge. Our interest is in alternating odd cycles in H, is that they are associated with augmenting path in the original graphics. We are characterizing these odd cycle by the fact that they always donot have atleast on B edge. So, how to be prove this claim? Let us say, there is no B edge on the cycle, suppose we call the cycle. Let X be an alternating odd cycle in H, which contains no B type edge. So, suppose there is 1 such alternating odd cycle containing no B.

Now, let us brake this X into pieces namely P 1 e 1, P 2 e 2, P 3 e 3, where the edges associated with the tree edges are type C edges are in this P and even is a type A edge. Now, you have only either the tree edges namely E T edges which are these, or you have type A edges or you have type C. So, these contain only tree edges and type C edges and this is a type A edge. Once again these are either tree edges or A edges and so on.

Now to begin with, P 1 starts at S. Starting from S there is a no type A edge, there is type B edge and there is no type C edge. So, we descending along 1 of these edges. So, P 1 has atleast 1 edge. So, when we proceed along P 1, what we notice is that this must continue to descend because these start with in unmatched edge, then we must go along the matching edge. We are not allowed to go along type B, so we cannot go this way. We got to go along with it now.

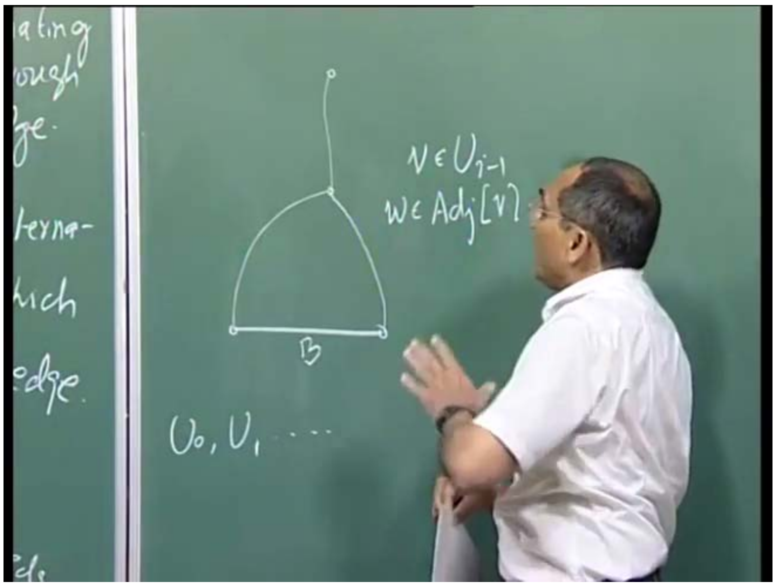
It is possible that we may have gone like this. We may have taken as C edge. Now, at some point we go with along one of the A edges. Now, we notice that an A edge is always connected to an odd level and the other end can be at any level. It could be either immediately the next one or any lower level. So, in case the P 1 part as at odd level and then if it takes a descending A, then what we notice is we will have an unmatched edge followed by an unmatched edge. That is not allowed in a alternating path. Hence, it has to be the tower strum through which must have from an even level to an odd level.

That can happen either from U 2 to U 1, or it could be from U 4 to U 1, if there is an edge like that. So, assent the rising can happen through an E x. Starting from an even level which will go back through an odd level. But, the moment we reach here, then make edge has to be a matching edge, so we start descending. So, next P 2 must start from a matching edge and it should continue to this descend once again. If there is a type E edge, then again it should P 2 should terminate at even level and our E 2 must reach the path through an odd level going up.

Now, this process can go on, but what we notice is that at no point any A edge terminates in S. As it is impossible for this path to reach S. So, this is obvious that unless a B edge is involved in the path this circuit cannot be completed. Hence, what we have seen here is that every alternating odd cycle must pass through a B edge. Now, for example, in this case this is a possible alternating odd cycle and so is this.

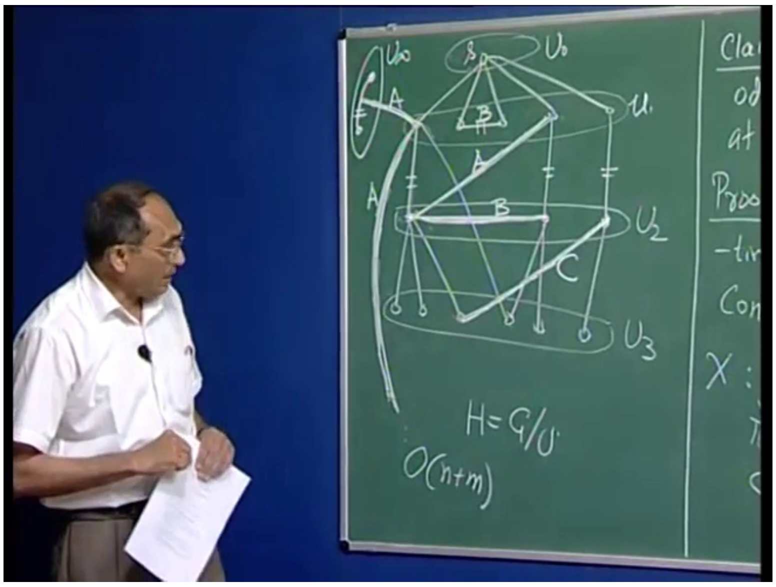
So, if we want to detect alternating odd cycle, then we can begin with in a B edge and try to see if there is a cycle which can complete the path. So, let us just take this situation. Suppose, I have this thing and I start rising from here, it so happens that we have immediately vertex A, then obviously we have found an alternating path. The same way, this tree in this case because here is a B edge, I can start rising from the tree path the tree edges. I will reach here in this case. I will reach here, but this is a tree and it is possible that as we rise along the tree edges, we may meet at some other vertex.

(Refer Slide Time: 48:44)



So in general, if we start from the edges from a B edge we will go along the tree edges and eventually find it common in vertex in the tree. From here of course, there is a path to S. Of course, as such this not an alternating odd cycle, but this is a structure of which will see in the next class. Another comment is about edges B. Please not that out of them we have identified all the classes U 0, U 1 and so on. Then any edge between any pair of vertices of the same level is a B edge. To detect any B edge, all we have to do is take each level and check if there is an edge between any pair of its vertices. That will be the set of all the B edge. So, to compute the tree and the set of B edges, please note that this entire process is essentially a breakfast.

(Refer Slide Time: 50:26)



Hence, the computation of this tree takes only in order n plus m times, where m is number of edge and n is the number of vertices. Now, why we are searching in the algorithm when we are looking at the vertex V in a level U i minus 1? We were looking at vertex W. It is adjacent to V and there we where checking the level of this vertex. If the level of this vertex is i minus 1, then what we have detected is that there is an edge between V and W. Both of them are in the same level. So, while detecting itself, we can identify the type B edges. Hence, it does not take any extra cause to detect all the B edges in the graph.