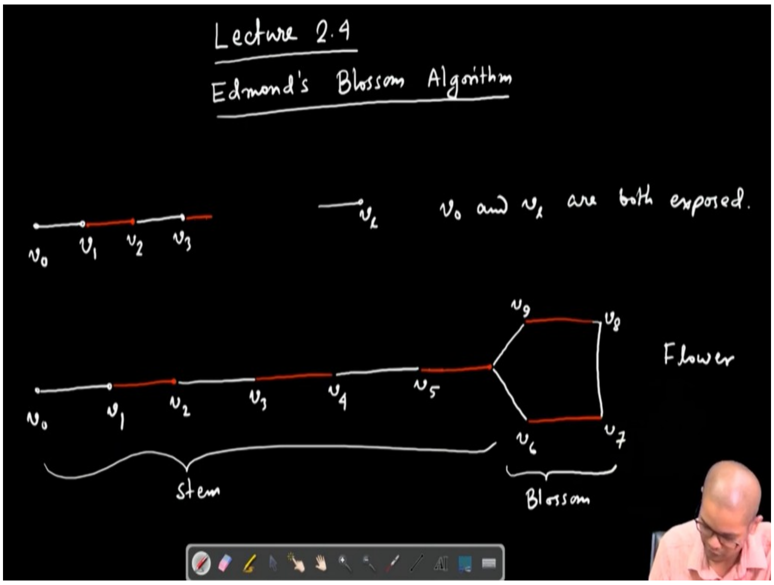
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No # 02
Lecture No # 09
Edmonds Blossom Algorithm

Welcome so in the last lecture we have seen how augmenting paths play a crucial role in computing maximum matching. In today's lecture we will see Edmond's Blossoms algorithm.

(Refer Slide Time: 00:40)



Let us recall that finding maximum matching boils down to finding an M augmenting path if m is the current matching because if there is not aim augmenting path then the current matching m is maximum matching. So how to find an aim augmenting path so let us take the natural algorithm. So let u or let us call it v o be an exposed vertex because augmenting path need to start from v o from some exposed vertex and it should end at an exposed vertex.

So if it has a neighbour which is also exposed then we have found an aim augmenting path it is an augmenting path it starts and end both at exposed vertex. So without loss of generality assumes that this is a not exposed so this is all its neighbours are matching. Let us pick any of its neighbour so this must be matching. So let us take the matching edge and if this let us call it v_2 if v_2 has any exposed neighbour then I have found an M augmenting path otherwise this search continues and so on.

Now 2 scenarios can happen one is that this search ends me at an exposed vertex that v_0 and v_l are both exposed then I have found an m augmenting path which I can use to augment and increase the current matching. Otherwise what can happen is that let us draw again $v_0, v_1, v_2, v_3, v_4, v_5, v_6$ and may be again this. So this sort of structure is called cloud so this process either this greedy process either will give me an m augmenting path or it will look back.

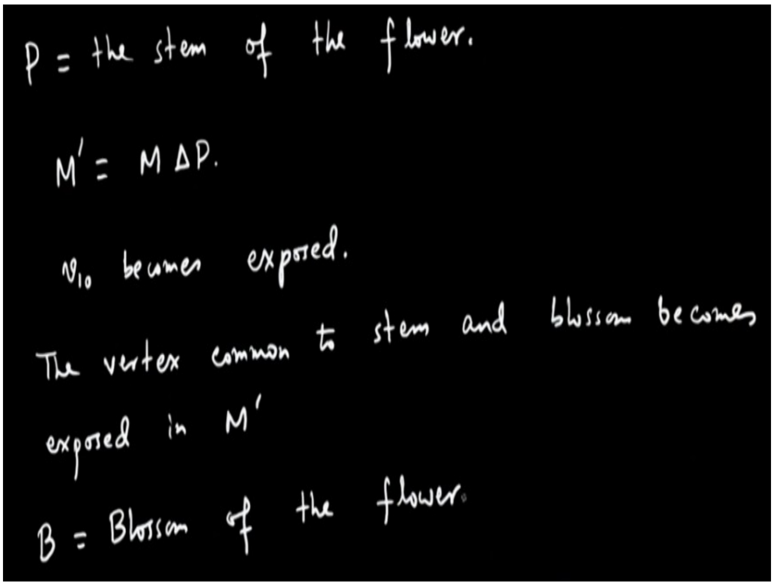
Because the number of vertices is finite and will get this sort of structure now this structure let us call this cycle part blossom and the earlier part v_3 and v_6 cannot be same. Because 2 matching edges cannot share end point so there should be the red vertex and anther matching is here. So v_6, v_7, v_8 again this was also fine v_3 and v_6 are different. So this stem and this blossom so whenever we obtain such a structure here is a beauty of Edmond's blossom algorithm is that.

We can use this structure to make progress and that is how first what we do is that I can ensure the vertex which is common to blossom and stem. So this is an even cycle and this is not useful so what we will do "FL" excellent "FL". So if we found an augmenting path that is good otherwise what we do is that again I start this search here is from v_0 then v_1 these are non-matching edge.

Then I have a matching edge let us call it v_2 and then again non-matching edge v_3 then matching edge v_4 then non-matching edge v_5 and so on. And we will have a structure which look like this so it is a matching edge and there is a say non-matching edge there is a matching edge say non-matching edge then I have a matching edge and then a cycle. So let us call them v_5, v_6, v_7, v_8, v_9 so if I have such a structure we if this path cycles then again I can make a process and that is the beauty of Edmond's Blossom algorithm.

I have found an augmenting path and such a structure is let us call it flower and it has 2 parts one is called blossom and the first part it is called stem both could be of arbitrary length, Now next what we do is that?

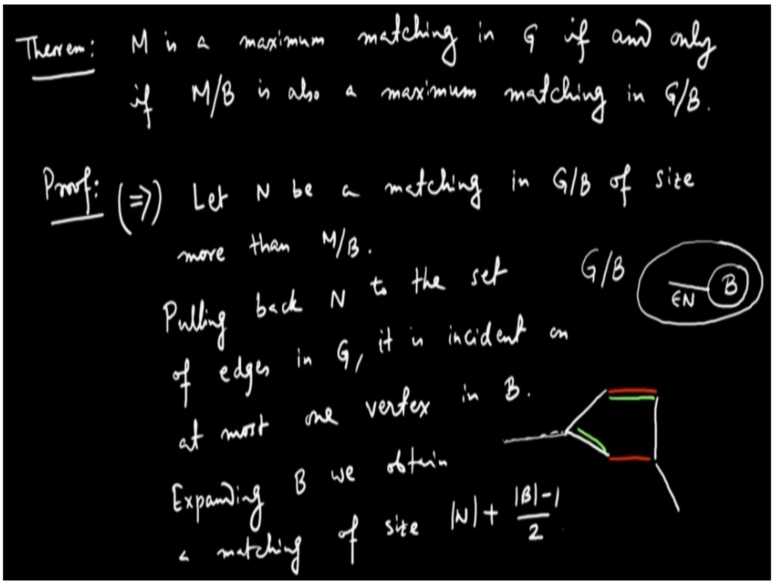
(Refer Slide Time: 09:07)



Whenever we have such a structure so let us call the steam P be the stem of the flower and we first exchange these matching edges with this stem we take symmetric difference so that this vertex is very important this called this is some important this vertex this green vertex become free. So how do you ensure that? We take new M define $M^{'}$ is M symmetric difference P that way this structure will now look like you know the first edge.

So this will be how this new matching will look like and it relives this edge this vertex v 10. So v 10 becomes exposed this stem and so made this way the vertex common to stem and blossom becomes exposed in $M^{'}$. And this is an odd cycle this is very important this is an odd cycle so we will see how we can find this odd cycles. Now once you found this object then what we do is that let p be the stem and B be the blossom of the flower.

(Refer Slide Time: 12:13)



So we show we claim that M is a maximum matching in G if and only if M/B what we do is that we shrink B we collapse all the vertices of B into one super vertex and let us call that graph mod B. And similarly M also all the vertices of the Blossoms is identified as one vertex let us call that M slash B if and only this is M/B is also maximum matching in G mod B.

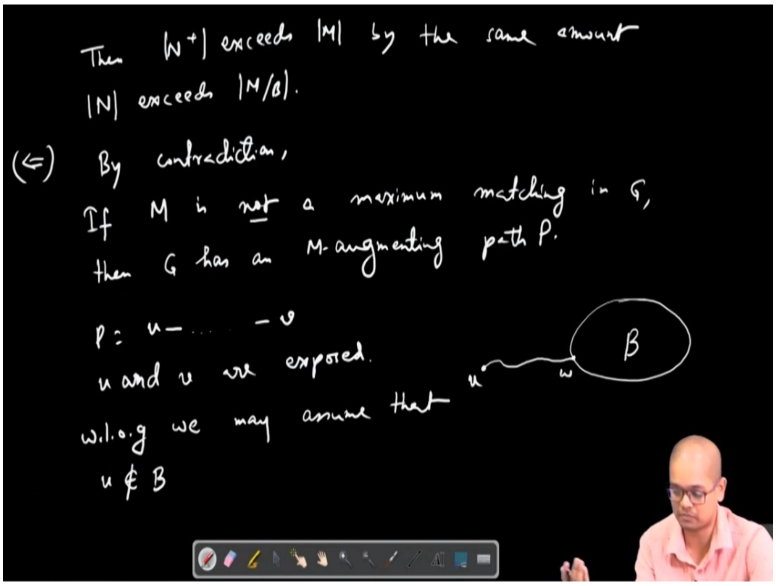
So proof so let me prove this direction first so suppose M is a maximum matching in G then we need to prove that M/B is also maximum matching in G/B. So suppose not so let N be a matching in G/B of size more than M/B. So you know in G/B how does this graph look like this graph you know all the vertices in B forms one vertex moreover this vertex B is an exposed vertex in G/B with respect to M this matching M.

So in G/B this is how is looks like and so if we; simply pull back N to the set of edges in G you see that in N can have at most one vertex from B. So it is so if N does not have any vertex is not incident on any vertex on B then those all the edges in N are also present in G. But if one vertex is if one of the edge in N is incident on B then recall that in B we had an exposed vertex and we can match that vertex we can rotate the matching.

So how does the structure look like no there was one exposed vertex and other was alternating like this. So if there is a vertex there is an H that belongs to N and it is incident on B then let us ask which vertex it is incident on if the edge is incident on the exposed vertex then I can simply use this edge because to this edge will belong to N. And this will be maximum matching on the other hand if it belongs if the H in N is instant on any other vertex what I can do is?

I can rotate I can pick other edges conveniently so that this becomes free so that then I will take this edge and this edge as my new matching. So in particular it is incident on at most one vertex in B. So hence expanding B we obtain matching of size of N + size of B – 1 by 2.

(Refer Slide Time: 18:33)



So then N + exceeds M by the same amount N exceeds M by B so in particular if size of N exceeds size of M/B and size of N plus also exceeds size of M and thus M will not be a maximum matching in G. So let us prove other direction that means if M/B is maximum matching in G/B then M is a maximum matching in G. So again this is a proof by contradiction so if M is not maximum matching in G then G has an M of augmenting path.

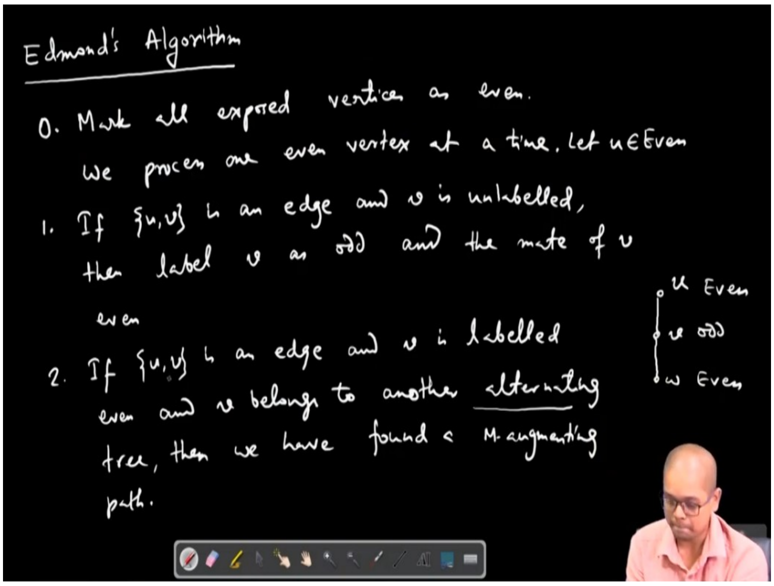
Now this was blossom B and this was the stem that stem is not necessary this blossom B that is more important and let us ask augmenting path. So let us call it p now how does this path interact mean? How does this path interact with this blossom in G? So here is this p and suppose this is the first path so this path p starts at say U and N Z B. So p look like maybe p = u dot v and u and v are exposed.

Now this blossom B has at exactly one exposed vertex so without loss of generality we may assume that u does not belong to the blossom it is not the exposed vertex in the blossom. So which is the first vertex so let us W be the first vertex in p which belongs to B. If no such W exists that means if the path entirely lies outside B then this is an augmenting path in augment M/B augmenting path in G/B also which will contradict that M/B is a maximum matching.4

(Refer Slide Time: 22:16)

So let W be the first vertex in p which belongs to B and Q be the part u to w. Q be the part of p which is at u to w then Q is also M/B augmenting path but this implies that M/B is not a maximum matching which is a contradiction. So this shows that whenever we get a blossom it is enough to find so what we get is that it is enough to find maximum matching in G/B.

(Refer Slide Time: 24:28)



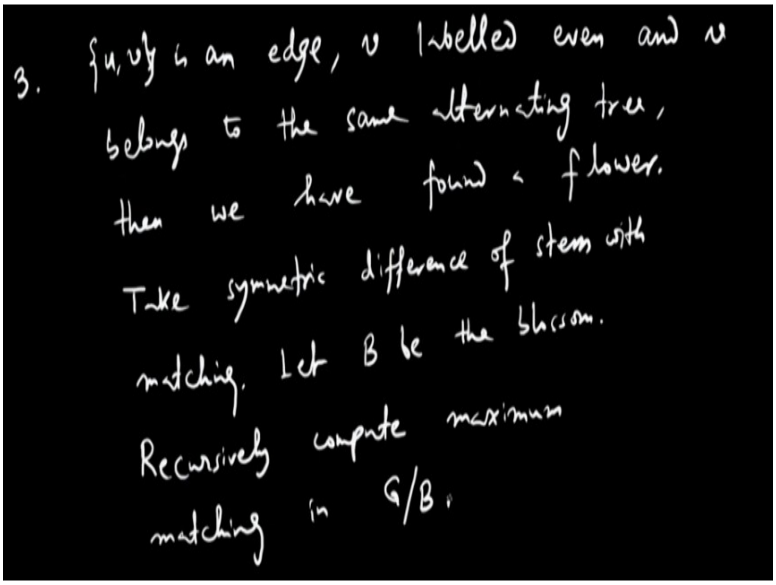
So now let me explain Edmond's algorithm so first assume step is mark all exposed vertices as even. In the process we will mark some more vertices even and some vertices or and some vertices will remain unlabeled. So will process one even vertex at a time so if u, v is an edge so let u be an even vertex let u belongs to even so every even vertex u I will run this procedure of u, v is an edge and v is unlabeled.

Then observe that we must be a matched vertex because we have initially marked all exposed vertices even. So what we do is that? We label v as odd and its and the mate of v the mate is the matching whoever which vertex v is matched with that is called the mate of v and the mate v even. So I have this vertex u and if there is an edge v this is v so this is even and if v is unlabeled I may label it odd.

And it is partner I label it even call it w call it label it even the second step is the other case is if u, v is an edge and v is labeled even. Then even and v belongs to another alternating tree so the tree we are constructing is called an alternating tree or even and so on. So if this v belongs to already marked labeled even and it belongs to an alternating tree then we have found up M augmenting path how?

Simply use this edge u to v u is exposed and v is labeled even so this v is w then you traverse from the root of this new this other alternating tree through this v and a use this edge u, u then you get an alternative and then you get an M augmenting path.

(Refer Slide Time: 29:35)



And the third case is u, v and edge is an edge and v labeled even if v is labeled or then again I have found an alternating edge an augmenting path and if v is labeled even and v belongs to the same alternating tree. Then we have found a flower so again what you do once we found a flower we will take symmetric difference of its stem and shrink the blossom. So take symmetric difference of stem with matching let B be the blossom recursively compute maximum matching in G/B. So this finishes the discussion of the algorithm so in the next lecture we will do the analysis.