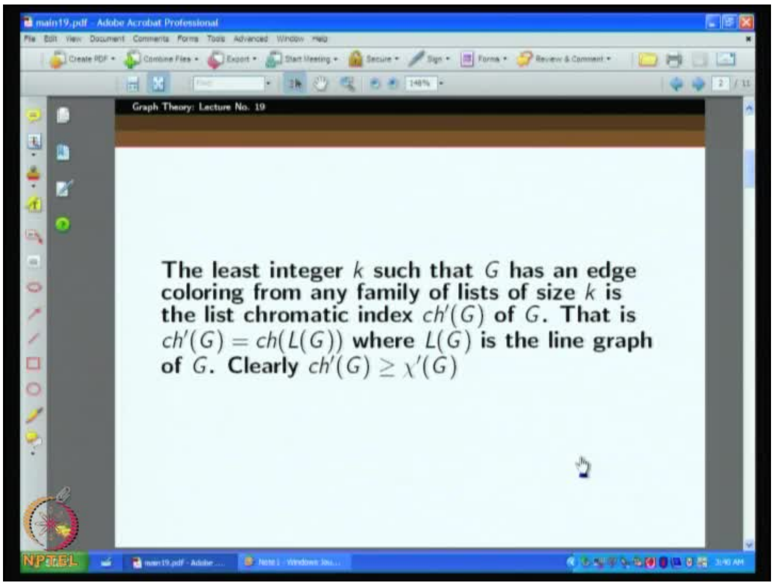
Graph Theory
Prof. L. Sunil Chandran
Computer Science and Automation
Indian Institute of Science, Bangalore

Module No. # 03
Lecture No. # 19
List Chromatic Index

Welcome to the nineteenth lecture of graph theory so in the last class we studied the list coloring for the vertex version in this class we will look at the edge version of list coloring problem so, the coming to the edge version so again we have we this time we have to color the edges of the graph and associated to each edge we have a list of colors and our intention is to color from this list in the the meaning is that the color that we give to an edge should come from its list in such a way that the coloring edge coloring becomes proper right

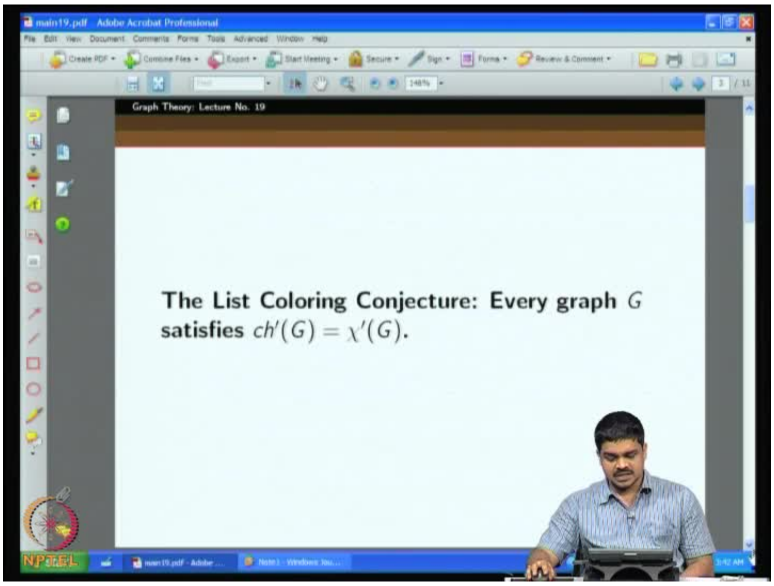
(Refer Slide Time: 01:17)



The least integer k such that g has an edge coloring from any family of list of size k is the list chromatic index of g other words so if we are ah if we know that each list has a length that means cardinality at least k then can we be sure that we can get a coloring for the edges proper coloring for the edges from the given list this number k the smallest integer k is that list chromatic index chi c h dash of g this is in fact you can also see this is a sub k is of the vertex version of list coloring because if we consider the line graph essentially the edges becomes the vertices and therefore, we can assume that in the line graph with each vertex you have associated a list of colors and we have to color the vertices of the line graph from the list associated with the each vertex

So essentially this c h dash of g is essentially the c h of l of g line graph of g the choose choice number of line graph of g so that is that is another way of looking at it so it is definitely a much specialized k is of the vertex version and clearly we know that the ah this list chromatic index is going to be greater than the chromatic index because if you choose all the list to be the same namely one two three up to chi dash then so definitely we know that we can color right get a proper coloring of the edges so in other words for a specialized set of list itself then we can definitely that get the chromatic index right so if we fix the list to be one to chi dash right for each edge then we get a proper coloring from that list

(Refer Slide Time: 03:33)



So therefore, the chromatic list chromatic index is definitely going to be greater than with seen but, the interesting thing is that that is the famous conjecture called the list coloring conjecture the in the exclusion these two numbers that chi dash and c h dash are going to be same that is what the conjecture says it is not they proved but, people believes of because there is not at least nobody has found out any example where the list chromatic index is greater than its actual chromatic index

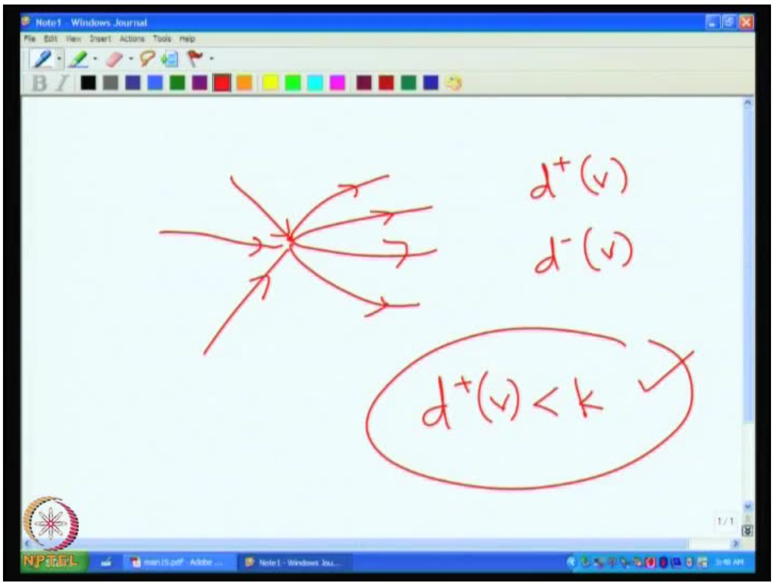
So the conjecture says every graph g satisfies chi the c h dash of g equal to chi dash of g so in this class we will show a special case of this conjecture namely for the bipartite graphs this conjecture is true this is what we know that when we consider the bipartite consider the bipartite version of the chromatic index problem we had seen that the vissings theorem says so delta delta plus one the maximum degree or maximum degree plus one is the only possible values for the chromatic index of a graph but, in the case of bipartite graph it is always delta the maximum degree

So here we are going to show that this same number will is going to be the list chromatic index also for the bipartite graphs in other words irrespective of the list whatever you put in the list as long as the lists each list associate with the edges of the bipartite graph have length cardinality at least delta then we are done we will be able to color from this list we will be able to give ah colors to the edges from its one each edge will get a color from its one list and the resulting color can be made proper right

This is this is what we are going to prove in this class in this this is our aim and now to show this thing we need some preparation so first of all we are going to look at a slight generalization of that degeneracy argument namely so we we had seen that if the vertices can be ordered in a certain way in such a way certain way means any vertex i th vertex if we look and you look at its lower numbered neighbors are depending on the way lowered numbered neighbors or higher numbered neighbors depending on the way

So one of the so let say lowered number neighbors if it is at most k all the time then if the number of colors is greater than k then we could vertex color the graph with k with ah sorry the number of colors is at least k plus one we could color it always or in other words if the higher number of higher neighbored number is always less than k then we can always color the graph with at most k colors this is what we had studied much earlier and in the list coloring case it told if each of our list where of cardinality at least k and any inform any vertex when we look we see the number of say lower numbered vertices is only at most k then we can color it using from the list right because of the because every time you we will keep coloring starting from v one v two v three onwards and when you reach v i.

(Refer Slide Time: 08:23)

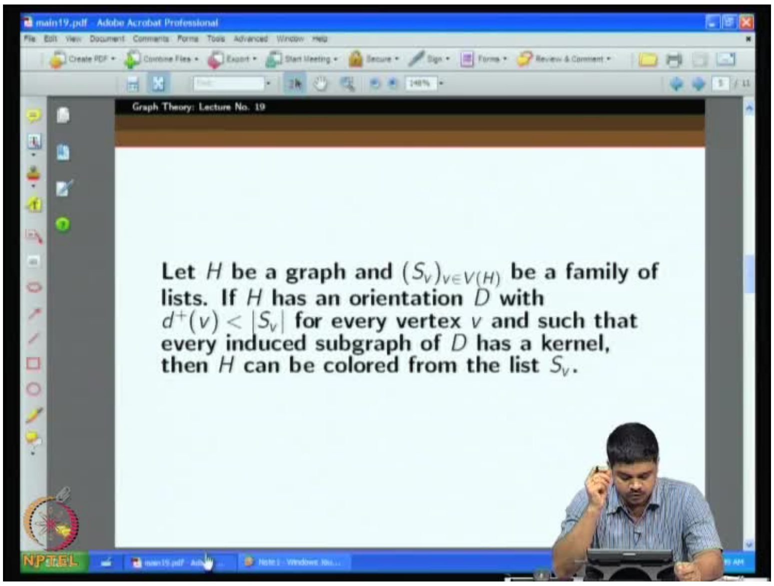


We know that the we have only used up less than k colors and the list as at least k colors there is one more list in one more color in the list we can color it using that extra color this simple idea if you want to generalized to a more i mean general case for instance see this is the way we can look at it for you can think of an ordering as way to give direction to the edges for whenever we see a v i v j edge we can direct an edge from the higher numbered vertex to the lower numbered vertex so it will give a direction and essentially at the i th vertex we will be asking how many edges are going out of the i th vertex ah going out in the sense that outgoing edges when after we give directions to the edges it will become outgoing edges right so the number of outgoing edges is at most if the number of outgoing edges for for for each vertex in directed graph we can see the some edges are like this out going some edges are incoming right

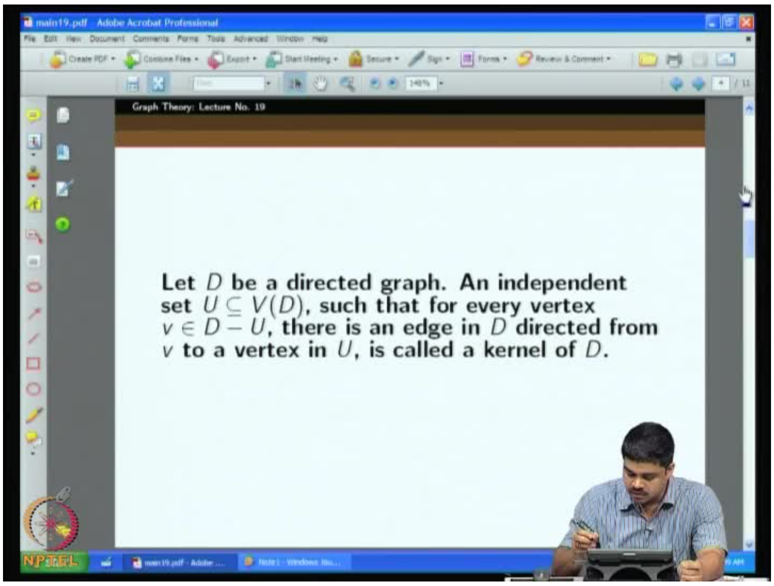
So usually the out degree that can be put as d plus of v and the in degree d minus of v how many edges are in incoming some many edges are outgoing if for each vertex d plus v is less than k then we can color with k colors is what we are saying in that case so assuming that our direction for edges for obtained looking at that the ordering of the vertices v one v two v three v v four up to v n we had numbered the vertices in such a way that the number of for any vertex i the number of neighbors which are numbered less than itself is at most or less than strictly less than k this was the ordering and after that we gave direction to the edges in a such a way that whenever we show a v i v j edge we gave the direction from the higher to lower number that means outgoing edges essentially counts the number of neighbors of a vertex whose value the index is less than itself right

So therefore, essentially it is like telling this d plus of v less than k d plus of v less than k then for each vertex then we can color it so this concept we want to generalize to the general (( )) why more general directed graphs so why more general directed graphs because this kind of directed graph for obtain from that is special ordering suppose we do not want to even consider about the ordering then the question is can we still tell something so this is this is what the next lemma considers ok

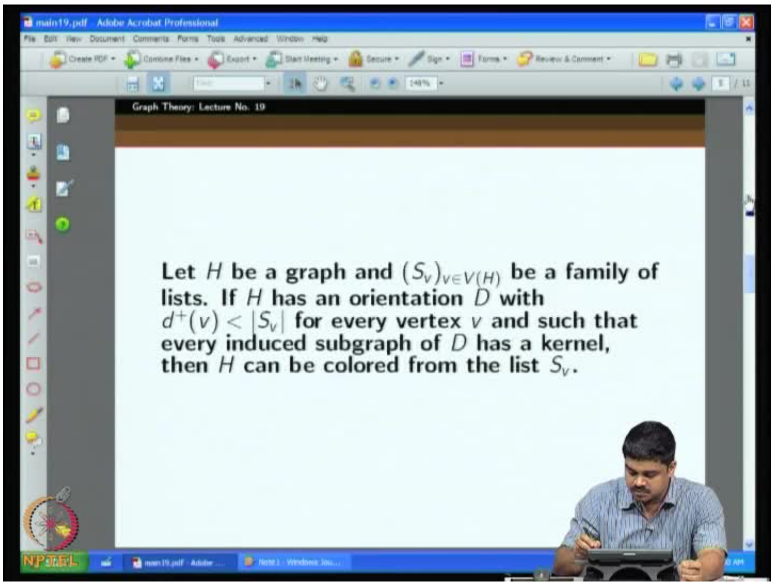
(Refer Slide Time: 10:37)



(Refer Slide Time: 11:14)

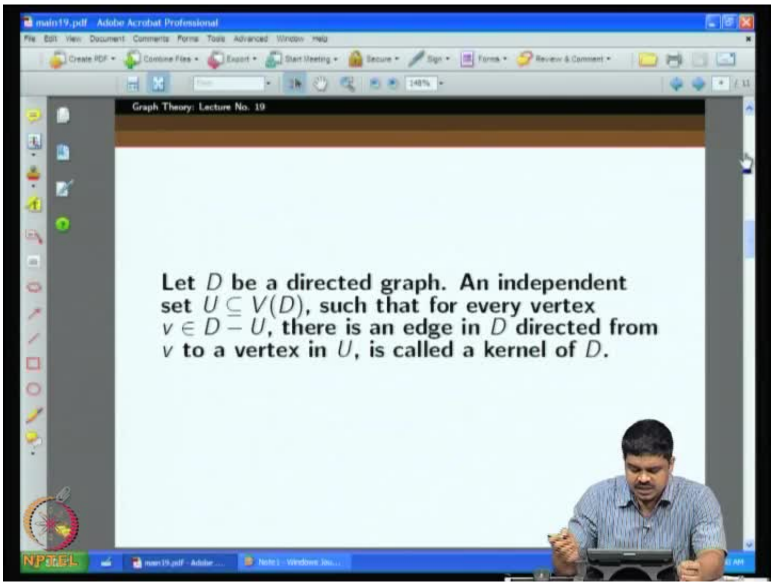


(Refer Slide Time: 11:20)

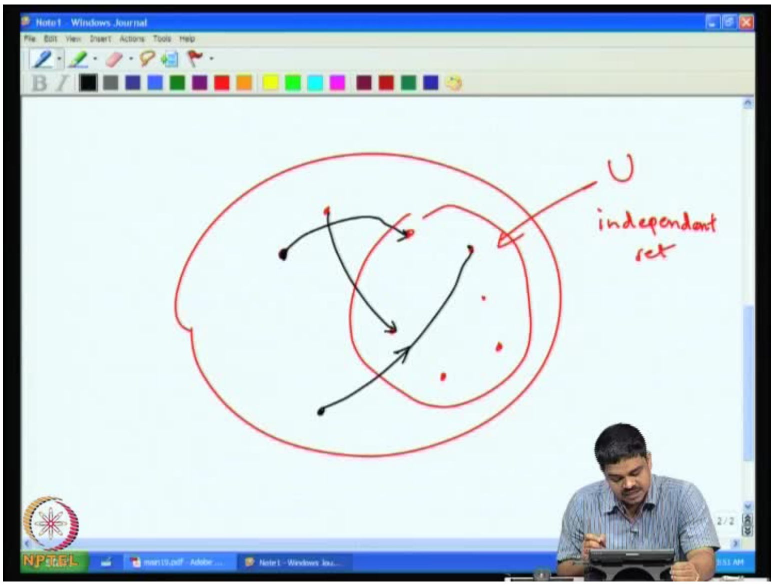


This is what the next lemma considers so, the lemma says let h be a graph and we have a family of list that means s v for s v is the list corresponding to v and if h has an orientation d that means if we can give direction to the edges of h in such a way that each out degree out degree of each vertex is less than the cardinality of its list s v for every vertex and such that every induced sub graph of d has a kernel so it is a concept here called kernel so this is we need to look at it so this is the key concept because we are talking about a concept called kernel which will capture our ability to color it with from the list if provided the list greater than (( )) so what is this kernel right

(Refer Slide Time: 11:34)

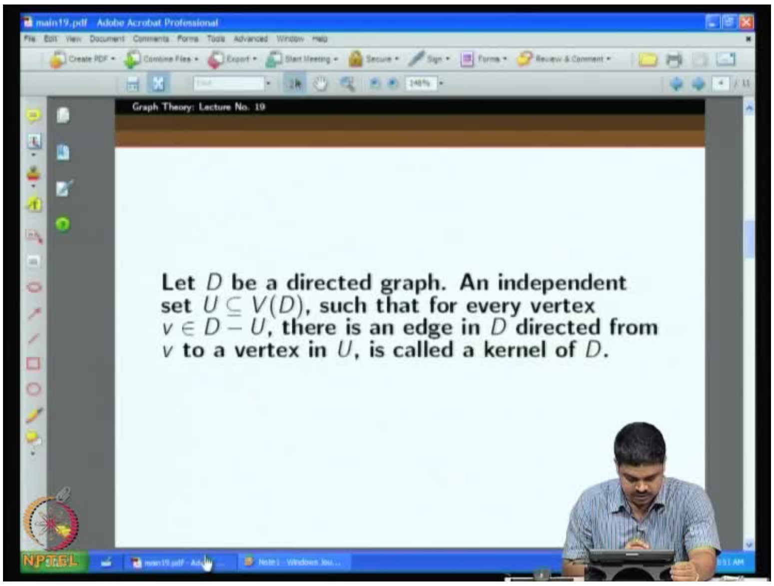


(Refer Slide Time: 11:54)

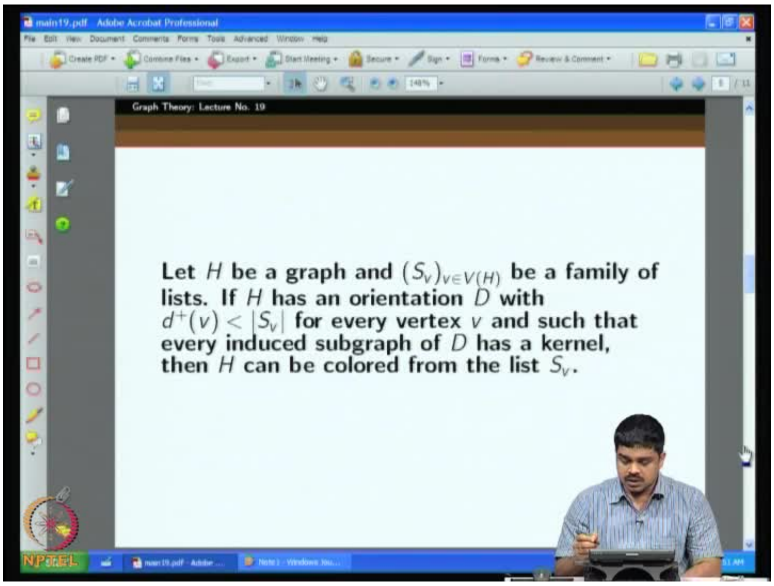


The let d be a directed graph an independent set u sub set of v of t such that for every vertex v element of d minus u there is an edge in d directed from v to a vertex in u is called a kernel of d it is something like this so you see you have a directed graph here all the edges of direction suppose you want you can find out some u here so is an independent set inside this so what should be the property of this u say independent independent set not only that if we take any vertex outside may be any vertex outside they should be an edge to some vertex inside u so you can find anything you should be able to find some edge going from that vertex in to u right ah

(Refer Slide Time: 12:49)

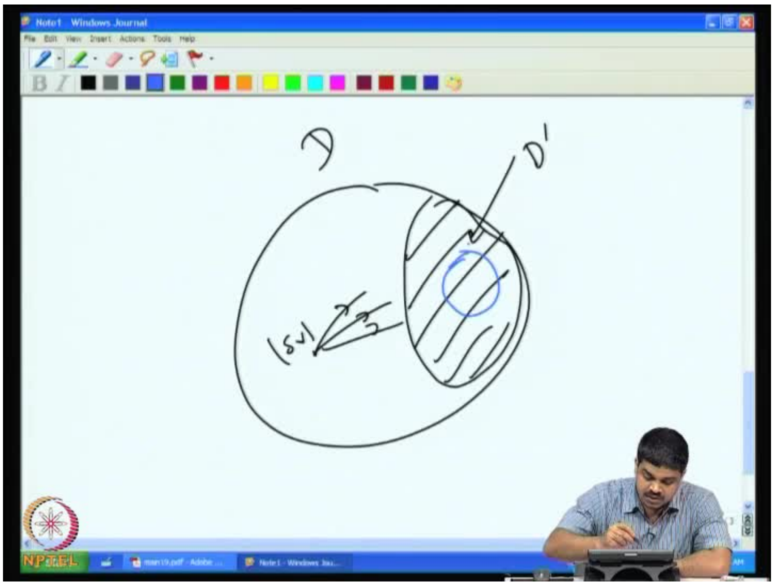


(Refer Slide Time: 13:07)



So any vertex outside u but, in d has to send an edge into u right any any vertex of u does not matter which vertex of u so such an independent set in d is called the kernel of d right this is what d be a direct so an independent vertex set use of set that for every vertex v element of v minus u there is an edge in d directed from v to a vertex in u is called a kernel of d now again coming back to our theorem so suppose h is a graph and you are given some list for the vertices of h that means each vertex has its own list

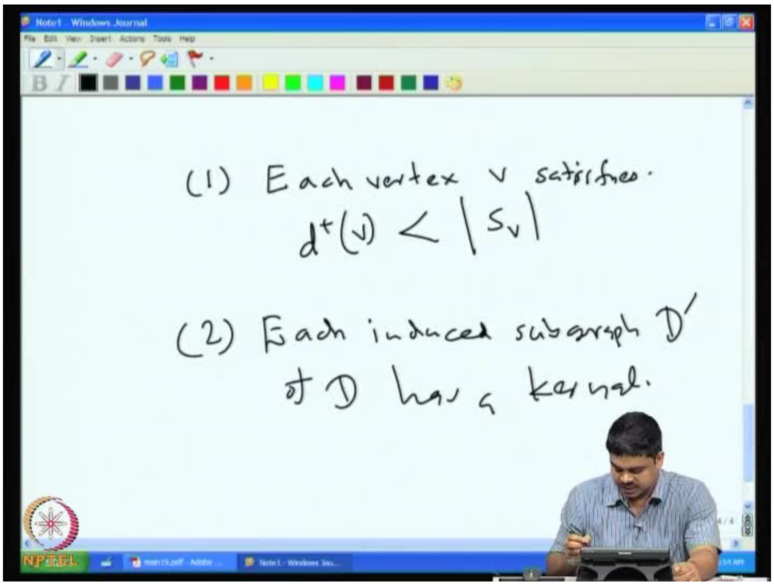
(Refer Slide Time: 14:02)



(Refer Slide Time: 14:34)

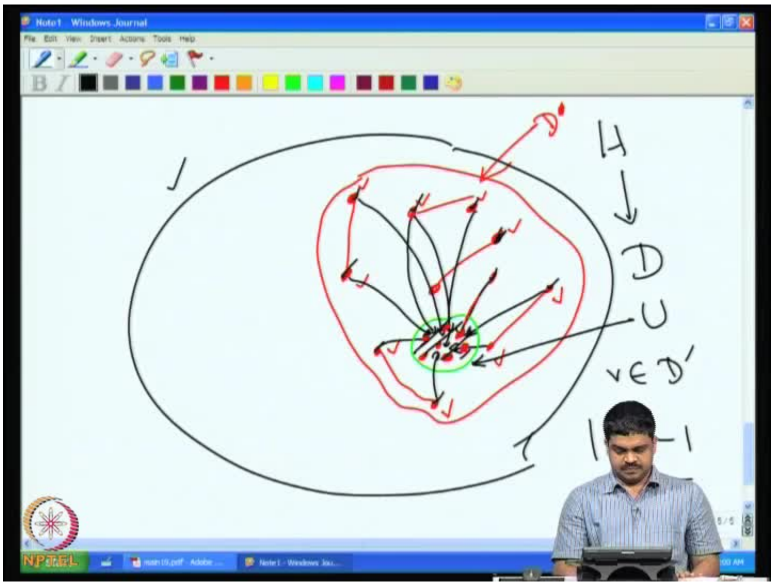
We call its v the list of v now we say if h has an orientation d with d plus v less than s v if h has an orientation d with d plus v less than s v cardinality of s v for every vertex v that means the out degree of each vertex has to be less than the cardinality of its own list and such that every induced sub graph of d has a kernel then h can be colored from the list s v the condition is not only that the out degree for each vertex the out degree has to be less than the cardinality of its list but, also that if we take any induce sub graph suppose i take this induce sub graph of it right so this is some induce sub graph we can call it d dash this is d now you should be able to find a kernel here some independent set inside here right something here such that such that it is a kernel that means every vertex of this d dash should send an edge in to this it should happen for every possible d dash induced sub graph d dash of d

(Refer Slide Time: 15:02)



Then we say that then then we we are sure we can be sure that the graph can be colored from the list given to each vertex these are the two two conditions one condition is one condition is each vertex v satisfies out degree of it is less than the cardinality of its list second condition is each induced sub graph d dash of d has a kernel each induce sub graph d dash of d has a kernel these two condition summit then we are sure that edge we remember d was directed graph created from edge so we can color edge using list the given list of the vertices of h right

(Refer Slide Time: 16:12)



How do we prove it to prove this so what we first do is to pick up a color so let say we can collect this this is the graph we gray we gave some edges this is edge first we created d from that by giving directions to the edges and it so happens that that condition through condition made for this thing now we picked up a color maybe we can pick up the red color and we locate the red vertices in it right sorry not red vertices essentially we haven't the color we are going to demonstrate how to color so what we have done is we considered the vertices such that the red color is present in the list is not that once you pick up a color red it need not be present in all the list right some vertices may be there in whose list the this red color may not be present we picked up those vertices in such that they list contain v so this subset sub graph so where these are all let say these are all the vertices whose list contain red

So we call it d dash now you know that the second property says d dash should have a kernel right let us call this kernel here small kernel is there for this thing these are definitely these are also colored this also vertices inside this also we have vertices whose list contain red color so because this an independent set we can give them all the same color what we are planning to do is to give red color to each of those vertices see these are essentially colored red now these vertices have only red in their list right these are just colored we just took the color red from the list and colored them because they have this red color in their list now but, we would not why cannot we color each of the vertices here because it is possible that there is a conflict there can be edges like this right

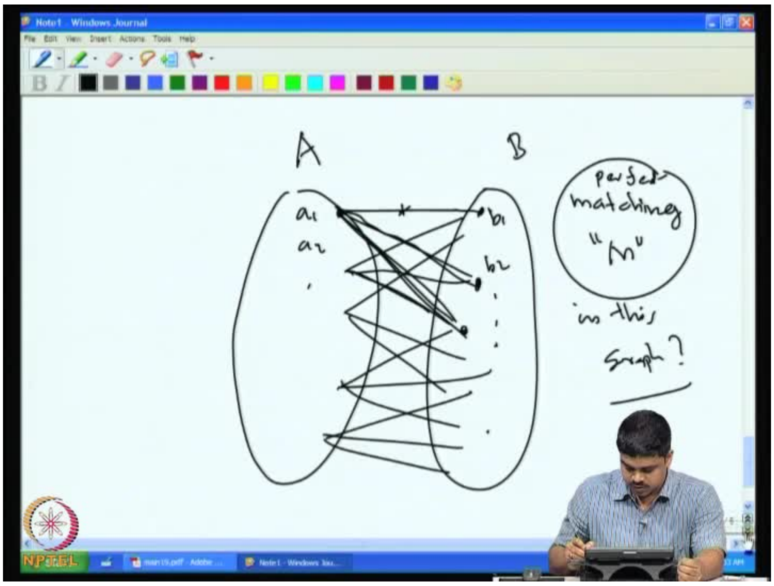
Because there are not independent set we can only color the vertices of an independent set with the same color otherwise they can be conflict so now what we are going to do is because this is the kernel you know every vertex here has an edge into this like this some edges coming into this independent set that is the importance of a kernel right everything is sending an edge in to this now you will delete the red color from the list of each of these vertices red color will be deleted red color will be deleted from the list of each of these vertices now what happen to the list of these vertices there list size when down by one for each v element of d dash so s v now has reduced to earlier s v minus one right

The earlier cardinality minus one one reduced because we removed the red color but, then we know that for each of those vertices one edges going in to the kernel namely this is the kernel you write this green colored cycle as the kernel u so once we colored this u and removed those vertices from the graph then these edges which are going in to them also will be disappearing so each of the vertices in whose list we had seen red color and removed them later then not only reduced the list size but, also reduced their out degree therefore, inductively we know that the out degree of each vertex is still less than the cardinality of the list there associated with the vertices whose list having reduced actually there out degree also is not changed therefore, nothing to worry about it though we have to only worry about those vertices whose list got changed because it is possible that then we reduce the list size it might have become equal to the out degree but, now out degree also reduced by one therefore, it will never happen right

Now we can remove safely this kernel which is colored red now and then work with the remaining lays red color itself is not there anymore in the system right now inductively take another color and do the same trick so by the time we finish of all the vertices every time we will finish of something right so we will we will end up coloring the entire graph from the given list this is the architect this is the simple enough argument but, this has generalized the the idea of considering an ordering such that the lower numbered neighbors are small right

To a more general directed graph this here we should not is that here this directed graph may be coming from need not be coming from an ordering need not be you can say you can have cycles and directed cycles it does not matter we we only ask for a kernel now the next thing is to show how the bipartite graphs can be list edge colored using chi dash of g colors typically delta colors chi dash of g for bipartite graph is delta the maximum degree colors right so but, before that we will just remember a useful theorem about stable matching’s this is so we can in fact we can do without it but, we will consider this for it is it is also useful theorem so this stable matching question is like this so here so you remember the matching question so here we have a graph bipartite graph let say right

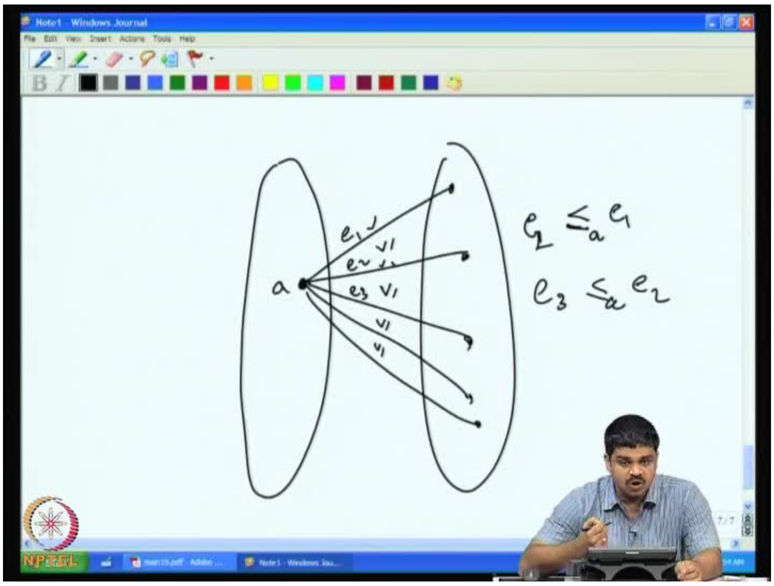
(Refer Slide Time: 22:44)



So the this is a side this is b side the a side vertices are usually written as a one a two like this b one b two b three see the there are some edges between this like this right some edges are there so the you remember in the initial classes we are consider the matching question so the idea was to ask does there exist a matching m in this graph right and so we usually we ask what is the now perfect matching in fact sometimes we asked about perfect matching but, if they perfect matching was not there we were interested in maximum matching’s.

What is the cardinality of the maximum matching can you find the maximum matching such a thing but, then typically in practical situations it not necessarily the maximum matching that matters it is possible that each member in the a side may have preference over the its partner for instance among so in general you can probably match any of the neighbors of a to it but, then a may say that i would rather prefer to be matched with b a one may say that i would be a rather i would rather like to be matched with b two then b one if there is a chance i would like to get b two than b one right

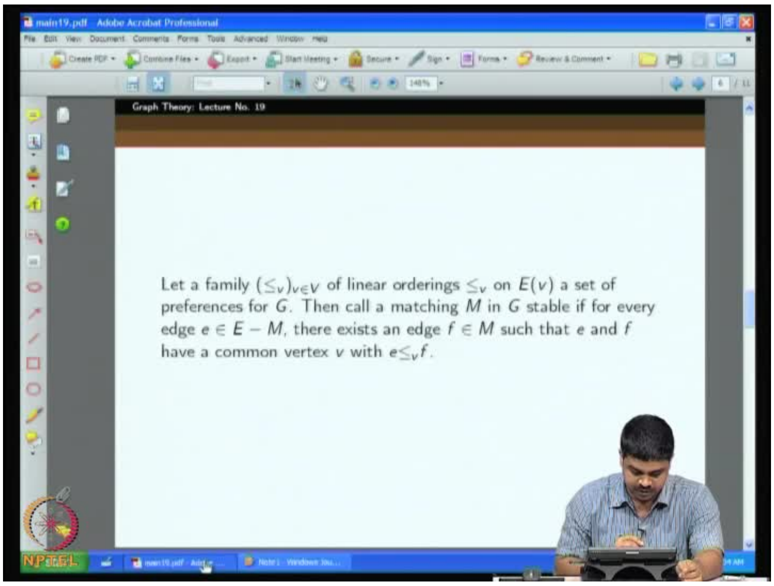
(Refer Slide Time: 24:32)



So so let me prove it in a nicer setting so here in other words if i take a vertex here this is a and it has this edges its neighbors of this typically a may spell out a preference a may prefer this edge in to be in the matching rather than this edge so may be something like this some ordering is required so we formally say that we have a preference order less than equal to with respect to a right this is the order

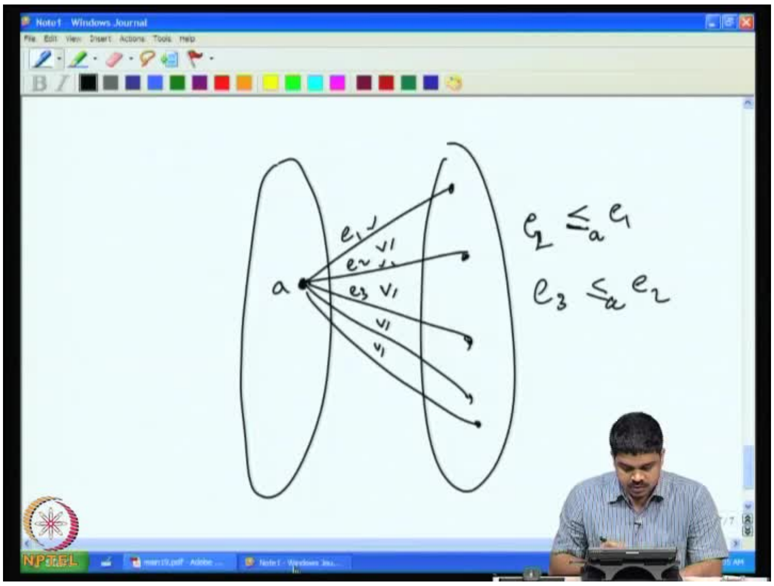
So so this is this order says so this is edge e one edge e two edge e three so may be e one is less than e one is e two is less than e one e three is less than equal to e two and so on total order among the edges so there is a preference among the edges the a can neatly order its preference from this is best for me then this is next best this is next best and so on right

(Refer Slide Time: 25:57)

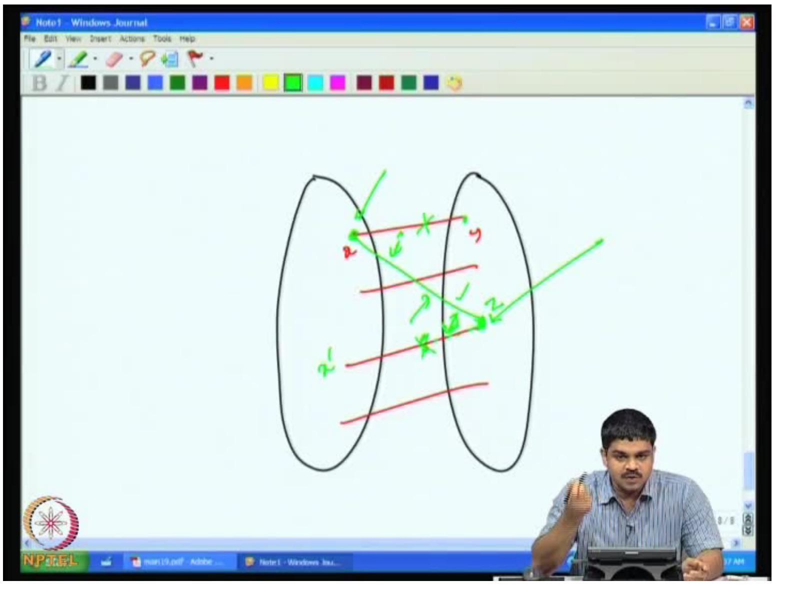


So not only each vertex both on a side and b side may give the preference of that shots this is called the list of preferences this is called the list of preferences so let a family of of linear orderings on e of v is the set of preferences for g then column matching m so we can consider a matching with respect to these preferences will say that the matching is stable e for every edge which is not in the matching which is outside the matching there exist an edge f in the matching such that e or f have a common vertex with e less than equal to f right

(Refer Slide Time: 26:44)



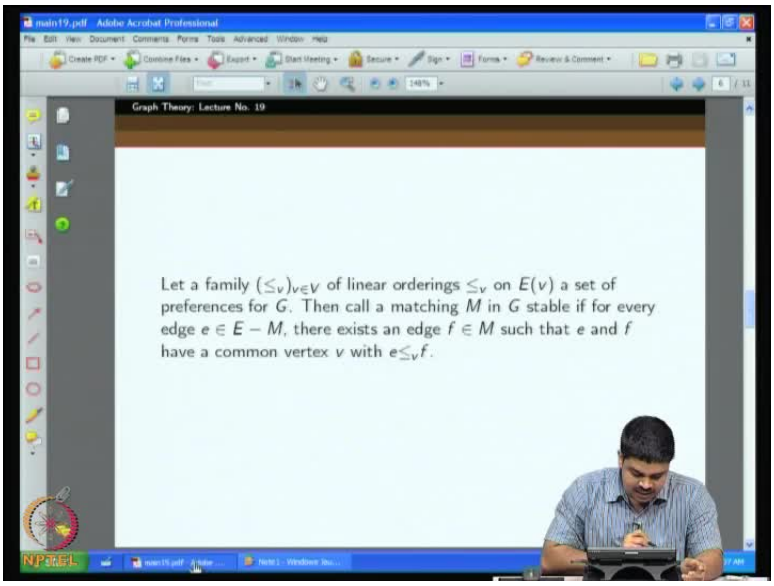
(Refer Slide Time: 26:50)



With respect to the ordering of that so the or in a common vertex that which vertex prefers the matching edge than the ah outside edge or in other words you can understand why it is called stable matching so let say suppose this is our we say that this this is our suppose this is the matching so it is so happens that here this is x this is y and suppose it happens that there is an edge here and this is z sorry edge here and this z has a higher preference for this green edge than this red edge than this red edge this green is preferred by z then this red edge similarly, this x prefers this green edge than the red edge suppose x is a human being and z is a another human being looking for the partner so if this happens then is very natural that x would give up this y and go with z isn't it or similarly, set will give up with its current partner say x dash and go with this thing so they will pair them all because both of them things that this other person is better and then there an partner is not good for him so they they would selfishly give up the current partner and match with each other

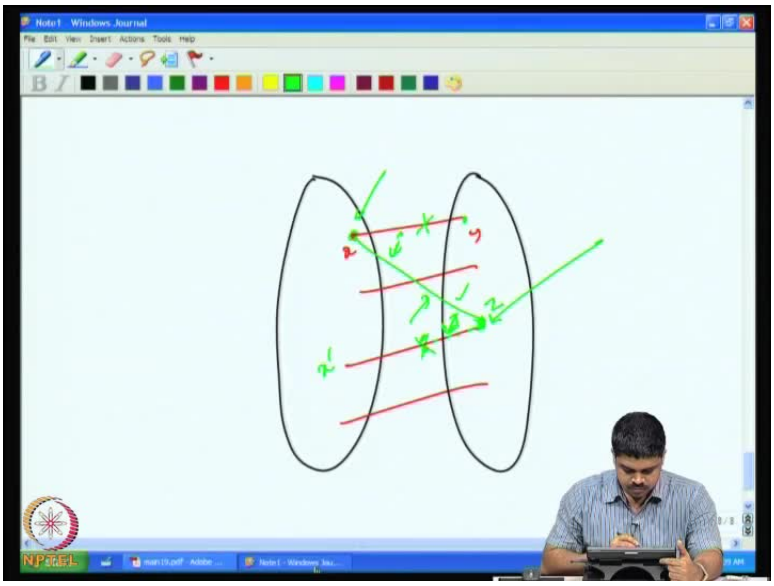
So therefore, we say that this current matching is not stable the stability is not there so when this the stability come stability come like this for instance x looks x things that green is better this green edge is better x wants to go go with the with set because we finds set as a better partner than y but, then set things its current partner is better than ah this green edge this red edge is better than this in that case set will not be convinced to give up its current partner and go that definitely z will not do any compromise or any scarifies for this (( )) of x we does not get any any advantage and doing that therefore, the matching will say even if x wants to give up its partner and go with the other partner z

(Refer Slide Time: 29:09)



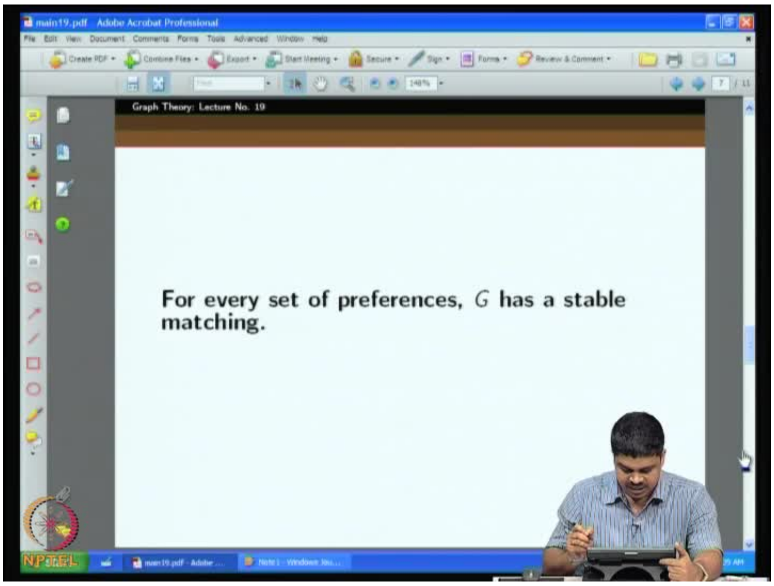
So z will not agree to that therefore, that way that the stability will come so that is what we are saying this stability comes from this fact so for every edge which is not outside which is not in the matching which outside the matching like that green edge so we should be able to find one of the end points of the green edge either the x side other x side like such that the math the that vertex then one of the end point says i prefer the current partner then going with the whichever the partner this outside edge gives me but, i would rather we would the current partner than what partner i would get if this outside edge geo part on the matching right that is why he resist that particular vertex resist the change right

(Refer Slide Time: 30:38)

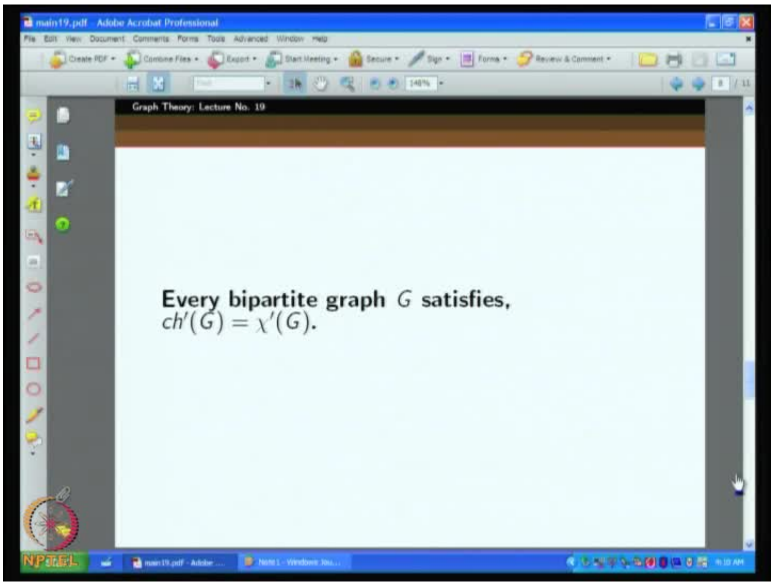


Because it does not give him its irrespective of the fact that other end point prefers to do the switching right because unless both the end points prefers a switching that will never happened right therefore, the stability comes from that particular fact that at least one of the end points resist the change so this is such a matching suppose a matching is such that no outside edge has a chance to swap or get inside the matching so by the the consensus of both the endpoints right so then we say that the matching is stable right the question now is yeah.

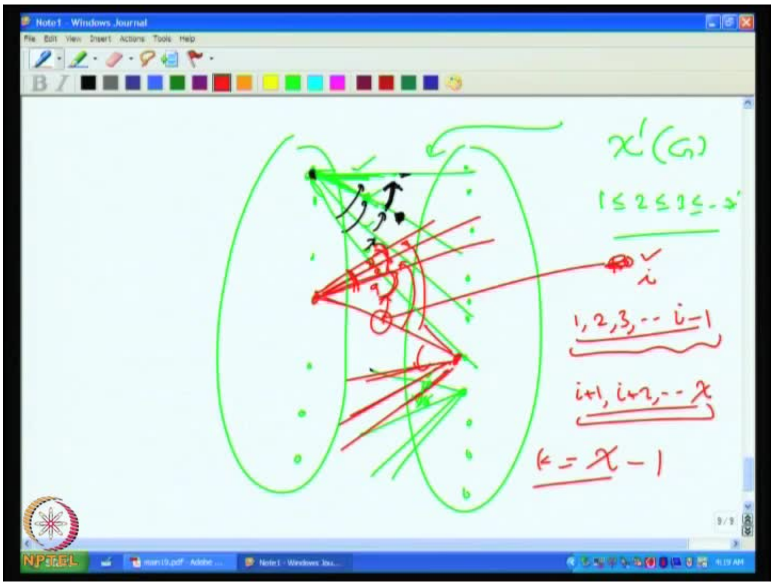
(Refer Slide Time: 30:47)



(Refer Slide Time: 31:44)



(Refer Slide Time: 32:10)



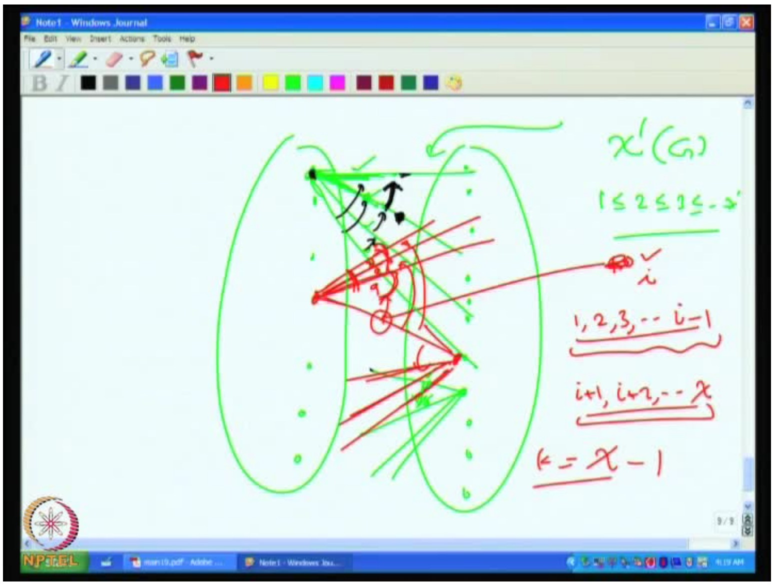
So thus so the we got interesting theorem of an interesting theorem of guideline shapely is that for every set of preferences you has a stable matching irrespective of how you how each vertex orders its neighbors the what whatever is there preference order he does not matter once you get the list of preferences for each vertex in the bipartite graph you do have a stable matching its possible to reach reach a stable matching there is no guarantee that this stable matching is is maximum matching it need not be a maximum matching but, stable matching will come that that is guarantee this theorem of guideline shapely we will look at the proof of this thing but, by the end of this thing why why did we do this thing it is because we are going to use this statement for the proof of our next theorem namely every bipartite graph is satisfies the list coloring conjecture namely chi the the choosing the chromatic list chromatic index of the bipartite graph is equal to the chromatic index of the bipartite graph therefore, we will need this theorem for proving it so we will we will see how will consider how we can go about proving the using the stable matching statement now you consider a bipartite graph so essentially we are going to prove that the line graph of the bipartite graph can be list vertex colored essentially that is want we want to prove so the because the edge are list chromatic index is essentially the the the list chromatic number of the line graph right ah

We can convert it to the vertex version so we can of case we can use the previous theorem about the the the ordering so,rry for the directed the kernel of the directed graph and if that statement which we just studied namely if the graph can be ordered sorry directed in such a edges of the graph can be directed in such a way that each out degree is less than the cardinality of the list corresponding list and also every induced sub graph has a kernel then we can list color it we are going to show that given a bipartite graph its line graph edges the edges of the line graph can be directed in such a way that ah such a way that these two properties are satisfied the two one of the two properties is that it is the list are always bigger than the list of bigger than the list of bigger than the the the out degree and for every induced sub graph has a kernel right

So let us look at the bipartite graph and because we are going to show that the any family of list such that each list is cardinality at least chi will do right at least chi will do right therefore, now it is a interesting proof we are not thinking about the coloring we are only trying to show that this line graph of this namely the the the line graph this bipartite graph can be oriented the edges of the line graph of the bipartite graph can be oriented in such a way that the two properties we mentioned earlier is satisfied then the rest of the things are satisfied right we will assume that the cardinality of each list is at least k therefore, we will only have to show that the out degree after the orientation for each vertex will be at most k k being the chromatic index of the graph and the induced sub graph of this resulting directed graph as a kernel for each each induced sub graph

Now we can take any particular so the this is the vertex right so this lets this we are vertex so we first edge color the bipartite graph that how many chi dash of g colors will be required so every vertex as a color now now the colors are to be ordered may be you can ordere the color one less than equal to two less than equal to three less than equal to the color can be ordered is up to chi dash of g right now will say this is the lowest color this is the second lowest color this is the third lowest color and so on now we can say that here this is the vertex in the line graph of this edge will become a vertex the edge will become a vertex they will be a adjacent right because there is a common touching point for them

(Refer Slide Time: 32:10)



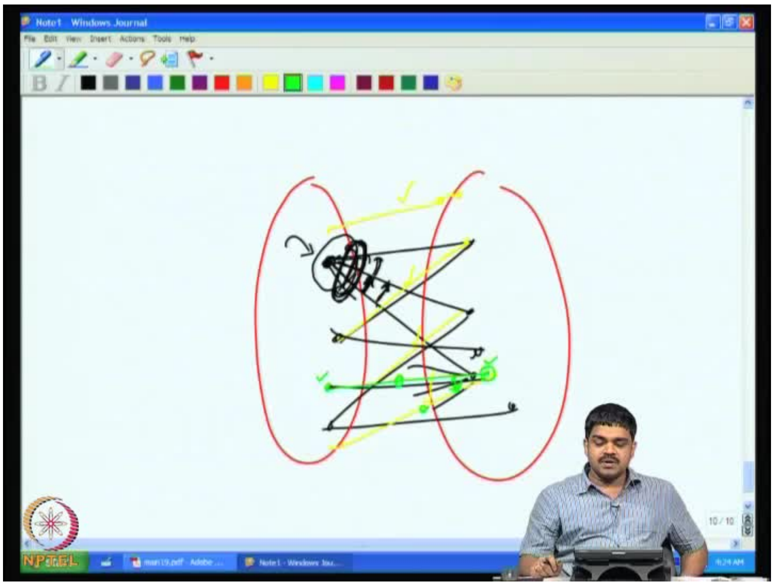
So then i will orient this edge the higher numbered color so like this you know for for in other words like this now in other words here when i consider this edge and this edge this has a higher color than this therefore, i will orient it from higher to lower looking at its color then the edge coloring each edge got a color so this color can be assigned to the vertices of the line graph and then when two line graphs in the line graph two vertices are adjacent we know the corresponding edges or the touching on some vertices two color those two colors are different we will and if they are from the a side so for instance that adjacent is from the a side here then we will always orient that edge from the higher numbered color to the lower numbered color on the other hand

If it was like this suppose the other way suppose it was like this it was adjacent to it is adjacent of the b side and send the line graph the corresponding two vertices are adjacent and these two colors we will orient we will look at the two colors and we will orient that edge from the lower numbered color to the higher numbered color this is the two differences so recall that in the line graph the vertices are actually the edges here and if in the line graph we see an edge between two vertices it means that the corresponding two edges are the original bipartite graph are adjacent if there adjacent on the a side the orientation is given from the higher color to the lower color and if there adjacent on the b side the orientation is given from the lower color to the higher color this is the orientation we we get an orientation of the edges of the line graph of the bipartite graph

Now the question is what is the out degree of each vertex of line graph that means corresponding to each edge of the bipartite graph let we can take an edge here this is an edge here so this is the vertex of the line graph is an edge of this graph how many neighbors of this we will be such that the edge going out of it in the line graph is directed towards the neighbor so,me of them may come from this side some of them come from this side right but, then the once which are coming from this side or definitely such that whatever suppose the color here is ten tenth color then this should be nine or eight or five or so lower colors only how many can be there only nine of them can be there for in other words if the color here is essentially i then the number of neighbors from this side to which the edge can go from this to this to seven this kind of edges is only at most i minus one one two three up to i minus one because only those many colors are smaller than this in similarly, from this side only i plus one i plus two up to chi can come right

So how many neighbors of this can be such that the edges going out of it so only this plus this many at most that many so that is chi minus one of them chi being the the vertex chromatic number we we are planning to give in each less at least chi colors right therefore, we know that the out degree of each vertex so each vertex of the line graph once each line of this bipartite graph is going to have going to be strictly less than the cardinality of the list cardinality of the list is going to be chi chi dash chi dash of the the chromatic index of the bipartite graph red is going to be less than that that is what we have shown but, the more important thing to show is that ah the the more important thing to show is that you can right

(Refer Slide Time: 41:08)



It take any induced sub graph and then you will see a kernel in that any induced sub graph of the the line graph let us any induced sub graph of the line graph means correspond to certain vertices those certain vertices are essentially some edges here we can just pick up those edges so essentially they will be some collection of edges here right

So now we we can discard all other edges so this is essentially the induced sub graph corresponding to that and then we ask do we have a kernel here so the we do have a kernel because we can translate it into a stable matching problem in the following way so you see the preference you know when i consider any vertex the the vertex between any edge of the vertex there is a ordering.

So you know there is an edge between we have put this is this this arrow is from this to this this arrow is from this to this this arrow is from this to this because in the line graph we have ordered the edges if you remember if you look at the line graph the edges which are touching this vertex or incident on this vertex will become a collection of vertices in the line graph they will become a clique there complete graph there so in the clique every edges added given a direction right and we know that the direction is a total order defines a total order because there is a preference here always from the highest colored edge will send all its edges outward to other vertices of the clique and then next turn will be sending its edges outward to all the vertices which are colored less than that

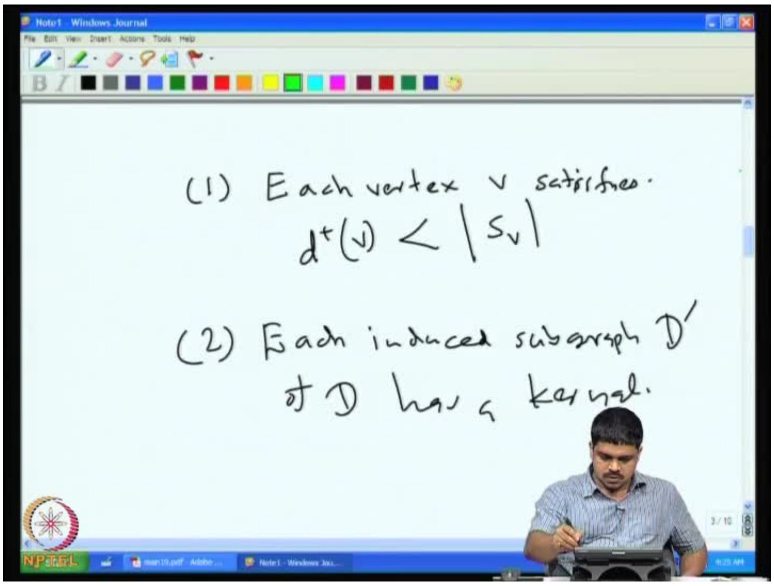
So that is the definitely in ordering with respect to the number the color numbers assigned on the this thing therefore, essentially if they essentially we can think that this vertex has this preference list on these edges say for instance if this this edge is sending his arrow to this edge that means this is more preferred than this the head of the arrow will indicate higher preference that means arrow is from e one to e two than e two is more preferred than e one that is what it is means similarly, for every vertex we have a preference list looking at that the b side of case it is the higher color higher numbered colors are more preferred well in the a side lower numbered colors are more preferred that is the only difference right

But then there is a preference raised does not matter how the preference claimed we we we know that there is indeed a list of preferences associated with each vertex but, then irrespective of what list of preferences each vertex maintains we know that there x is a stable matching with in that bipartite graph right so stable matching exist so that is the guideline shapely theorem which we just mentioned right

And this suppose say this is the this is such a stable matching right what is good about such a stable matching the the good thing about such a stable matching is that that can act as a kernel in the line graph the edges of that matching is indeed a independent set in the line graph and we can also argue that if you take any other edge it has to send at least one outward edge to the this independent set that to this stable matching why is it so because of case because it is stable see you take an edge for instance i can take consider this edge right

If this edge either here or here sorry this edge so you know at least one of the end points should resist that edge from getting in to the matching right why where placing some other matching edge so it does it means that at least one of the end points say this end point is incident with the matching edge and it prefers that matching edge that means the direction is from this to this so essentially in the direction when we look at the directions it will it this edge is directed towards this yellow edge and yellow edge is a part of the independent set of the line graph right.

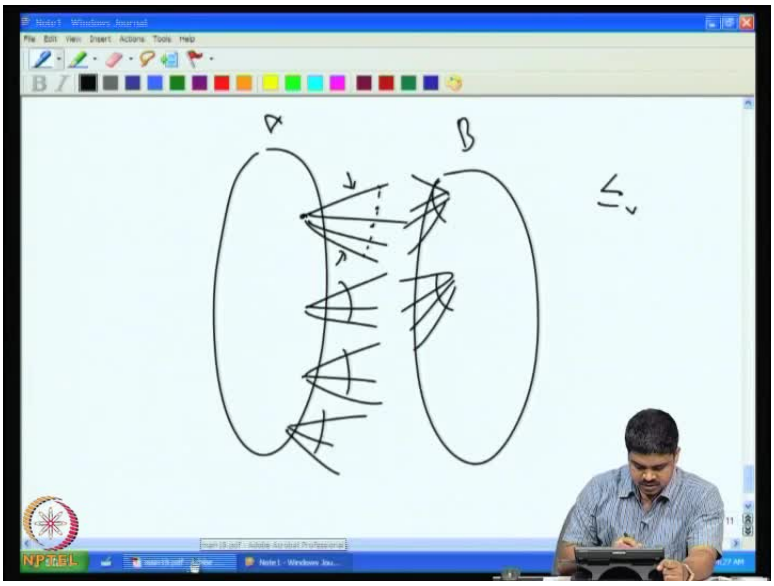
(Refer Slide Time: 46:27)



Therefore that is the arrow going from the corresponding vertex of this green edge to the corresponding vertex of the yellow edge and that means the yellow edges indeed form a kernel of the line graph so of the induced line line graph of this thing so this complete the proof so you not that the proof was by showing that when you take the line graph of the bipartite graph the two properties we described earlier is satisfied which are the two properties essentially maybe we can we can say that two properties these are the two properties both the we can we can we the clever thing was to find out the particular strategy to give orientation to the edges of the line graph means to consider whenever two edges of the bipartite graph incident on some vertex how to say that this is less than this right.

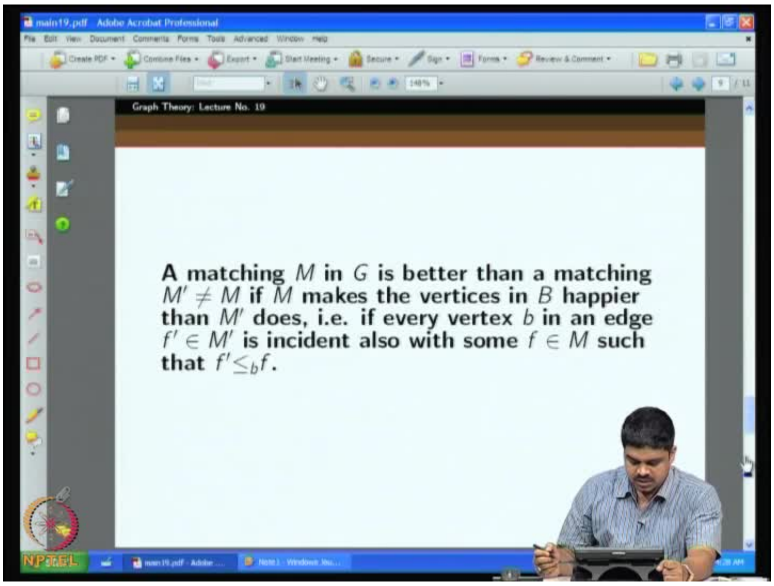
This edge is less than the other how to give a preference between them that is that is what we did right that is why we also want to the stable matching to finally, prove this thing therefore, by showing that in the corresponding directed graph which we cleverly made out of the underlying bipartite graph so and this and this two properties are satisfied we have proved that the list coloring conjecture is true for the the list colorings conjecture is true in the case of bipartite graphs that means its list chromatic index is equal to its list its chromatic number its chromatic index right edge coloring number this is what we achieve finally, before closing this class we have to complete the proof of the proof of the the guideline shapely theorem namely the stable matching theorem ah

(Refer Slide Time: 48:10)

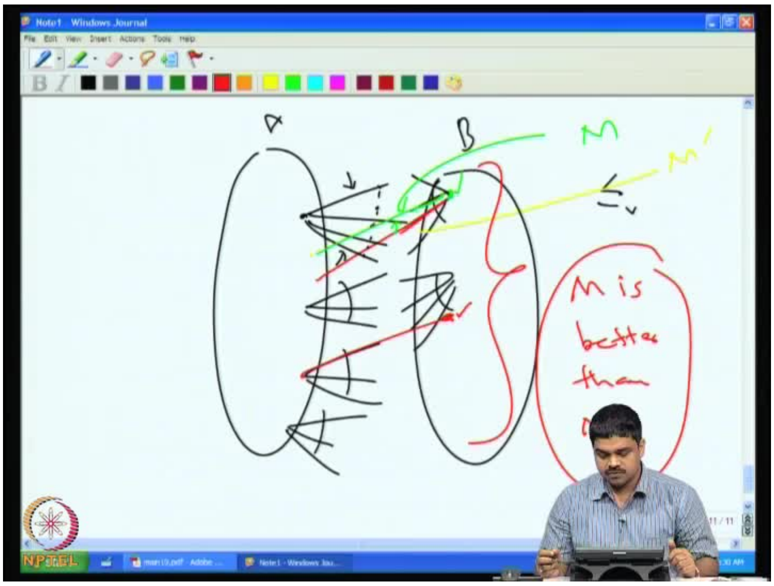


Stable marriage theorem we can say like that so the the statement was that you have a bipartite graph a and b and for each vertex there is a list there is ordering of the total ordering of the edges of it which is it preference that means there is the list of preferences corresponding to each vertex right each vertex has ordered its incident edges right there is an ordering for each of them is ordering we we wrote like this right for each v so that is the list of preferences for each vertex and

(Refer Slide Time: 49:19)



(Refer Slide Time: 50:02)

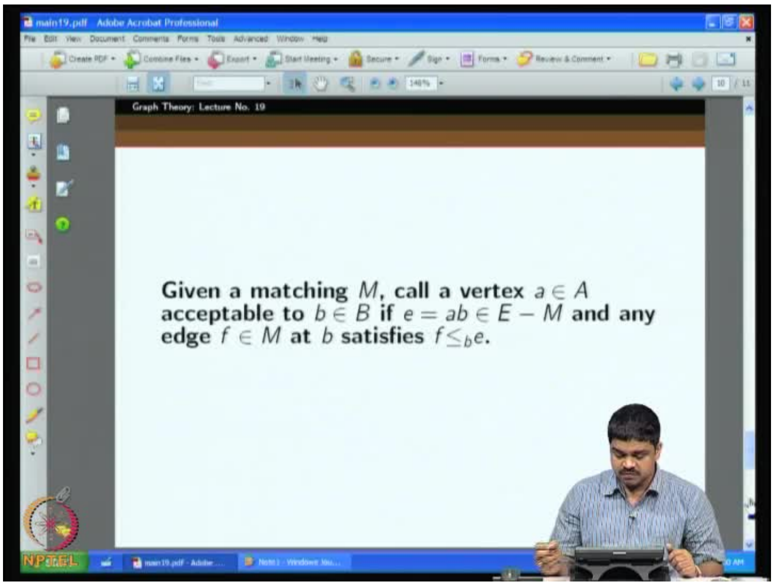


Now the idea is to show that there is the stable matching right the matching so to the the method is just a simple algorithm you start with an empty matching and we will show how to slowly grow it and get a stable matching before that to describe this thing we will need some couple of statements one is some some definitions so we we will say suppose there are two match consider a matching m in g and we want to say that matching is better than a matching m dash of g so will say that this m dash is better if are the vertices of b that m is the better matching happier matching for in other words the matching m makes the vertices in be happier than m dash in the sense that if every vertex b in an edge f dash of m is incident also with some edge f of m such that f dash is less than equal to in other words you can say that there are two matching's

So for instance if we take one red matching here and then we should so happen that this is red is m dash this is m dash it so happen that this should be a green matching so this this is a green matching is m so there should be another edge incident from the other matching such that this k prefer this one so the the green is preferred for this b side so every so it should not happen that here we have one here there is this vertex has no other green thing then definitely this vertex would rather he was not match that then he would prefer red right

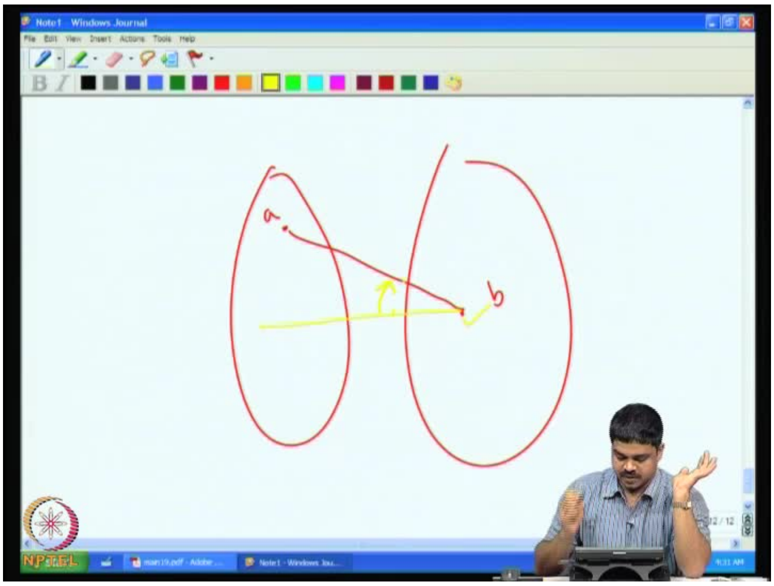
So every vertex on b side should get match by m also if it is match by m dash right should not happen that m dash matches it but, m does not match it the other way is now not only that the every vertex on the b side should say that i would rather prefer my matching my partner with respect to m rather than with m dash m dash then we say that for every vertex it happens then we say that m is better than m dash we should understand that there is a kind of asymmetric here

(Refer Slide Time: 51:37)



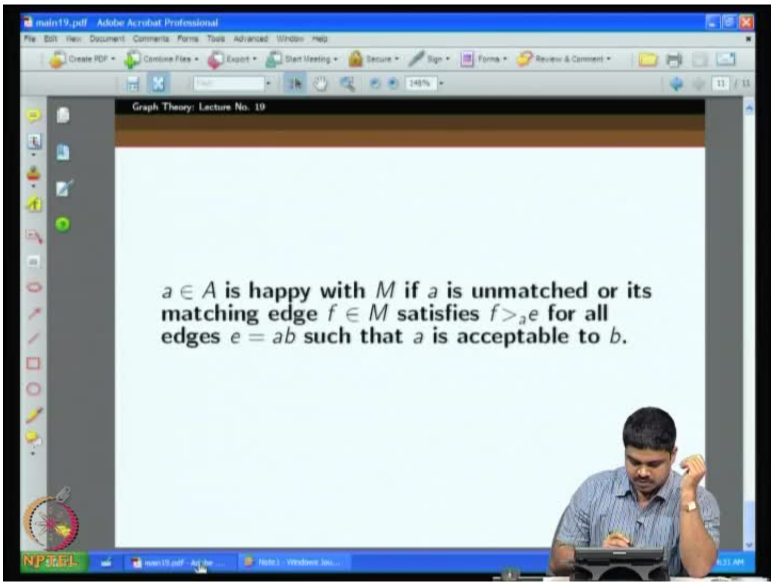
Because we are we saying we are saying that m is better than m dash essentially for the vertices on the b side essentially for the vertices on the b side therefore, that is one definition and then another definition we need is ah see when so you take a vertex on the a side we will say that the vertex a on the a side is acceptable to a vertex b when we consider that edge a b this should be an edge a b and it should not be in the matching it should not be the an already matching edge because anyway it is already match then what is there to be acceptable match so the only question is an outside vertex which is it is not a current partner can it replies this things

(Refer Slide Time: 52:34)



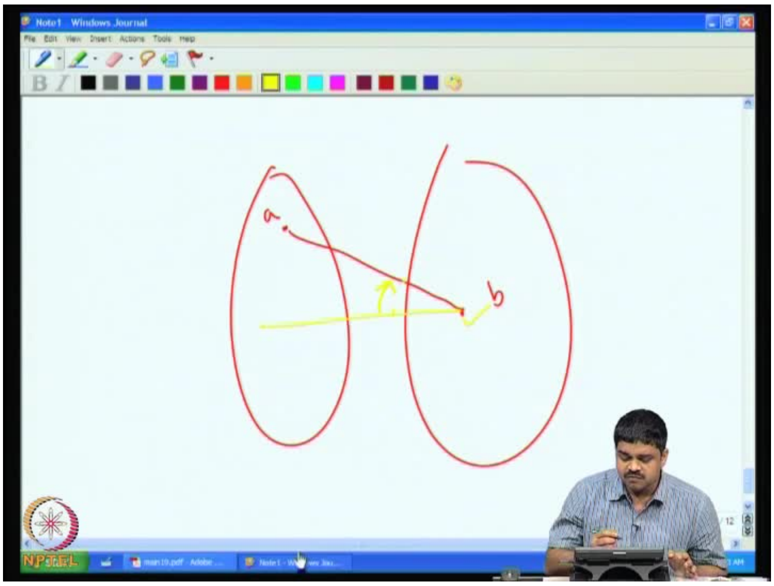
So it should be an non non-matched edge and the current partner of b we should prefer a to the current partner right so when so in other words we say that so here so here is an a and this b size a is acceptable to me for instance is edge is not currently in the matching and also when the b consider its matching edges say this is the matching edge so b should rather would like to go to this rather than this that means the this edge is more preferred then we say a a is acceptable to me in other words it likes b saying that i am ready to give up my current partner and come with a joint with a ah

(Refer Slide Time: 53:11)

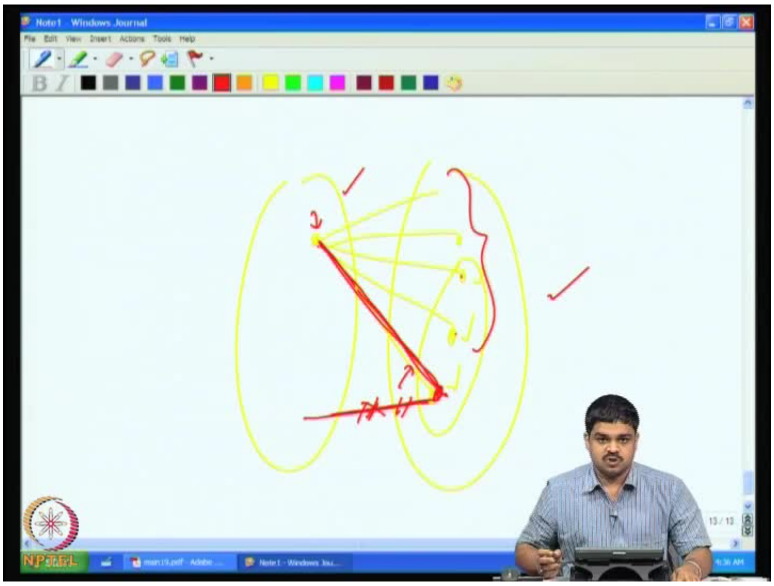


I am ready to make a my partner rather than the current partner right that is the acceptability the third property the next property is something call when do i say that a is happy i say that a is happy with m with the matching m right current matching i am happy for if i am un matched i am happy for i just (( )) problem i am a such along so i am happy if m match i would say that say i would look at my current partner and i will also look at my possible other partners which are the possible other partners the partner on the b side who are ready to accept me means there is no point looking at a partner who i am a like that partner better but, then that partner may not come with me because is he has a better partner then so in that case i have i have no hope but, rather i would look it the partners on the b side who are ready to accept me and among all those partners so when i consider those partners my current partner is better then i would say i am happy there are two situations one is i'm un matched than i am happy the second is if i'm matched among when i compare my current partner with a b side vertices who are ready to accept me to whom i am acceptable

(Refer Slide Time: 54:48)



(Refer Slide Time: 55:05)



My current partner should be better than all of them then i will be happy so the algorithm is to start with the happy matching namely the empty matching in the empty matching all the vertices on the a side are unmatched therefore, they are all happy next what it does is it will try to pick up an unmatched vertex a on the a side it will try to pick up an un matched vertices on the a side then it will look at all the neighbors of a on the b side to whom a is acceptable so suppose these vertices on the b side says that a is acceptable to me that means they prefer a than their current partner that is the meaning of that right with respect to the matching

Then among those things a would rather pick up the best one that means the highest preferred one so the then this matching edge will replies the edge which was already here right may be this we had an edge it may be possible that there is no b was also unmatched their is well and good but, if b is already match with some matching edge then i will drop it and put it the good thing is now we got a better matching than the previous one why is it show on the b side this everything all the b vertices are ah

So when you compare these two things only one changes happened there is one vertex has change the matching edge so and of case we we prefers the new edge therefore, we have move to a better matching now then again the you can see that the number of a the vertices on the a side are still all happy because nothing matches changed because a if earlier unmatched vertices now a new a has probably become matched a is happy so you can see that all the vertices have seeing their among the b partners which are to whom he is acceptable he has taken always the best one therefore, he is still happy therefore, we keep the this we maintained the property that on the a side everybody is happy all through the algorithm and finally, when we end up and we cannot pick up any a like that we have to stop that way will get a stable matching and we stop the class today and we will pick up some other topic in the next class thank you