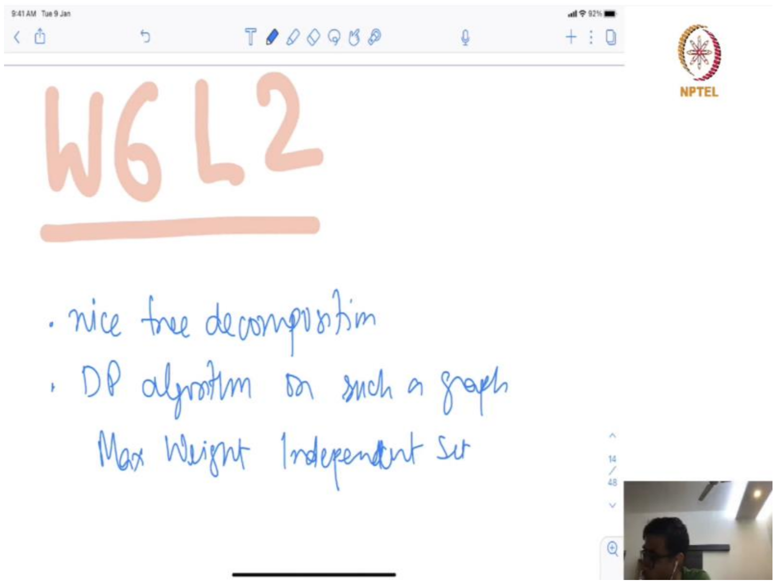
Parameterized Algorithms
Prof. Neeldhara Misra
Prof. Saket Saurabh
The Institute of Mathematical Science
Indian Institute of Technology, Gandhinagar

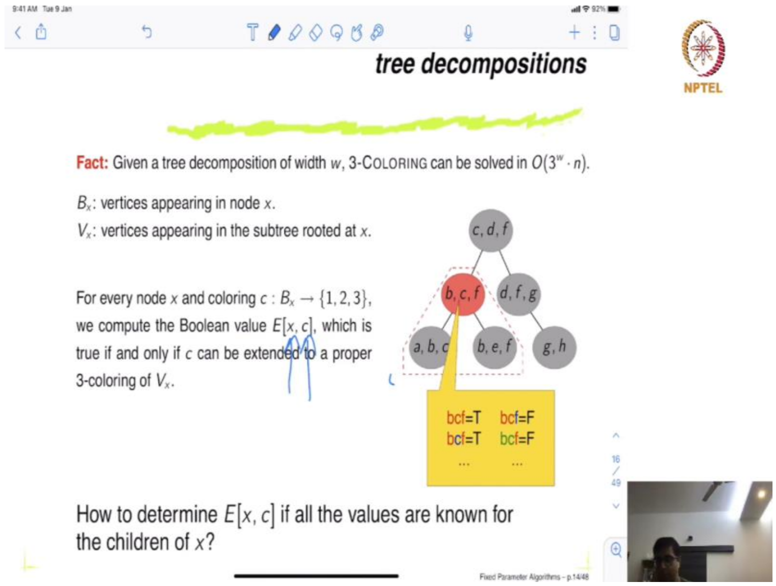
Lecture - 26
Dynamic Programming Algorithm Over Graphs of Bounded Treewidth

(Refer Slide Time: 00:16)



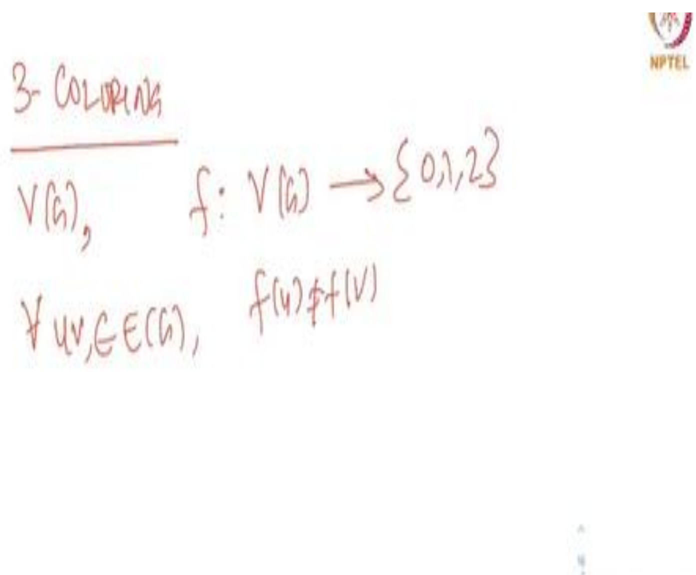
So, welcome to all of you to the second lecture of the big six width. So, in the first lecture we defined the notion of nice tree decomposition and we did our first dynamic programming algorithm on such a graph for maximum weight independent set. Now in this lecture we will see two three more examples of how to do dynamic programming algorithm on graphs of boundary tree width. So, just to get a hang of this new notion.

(Refer Slide Time: 01:13)



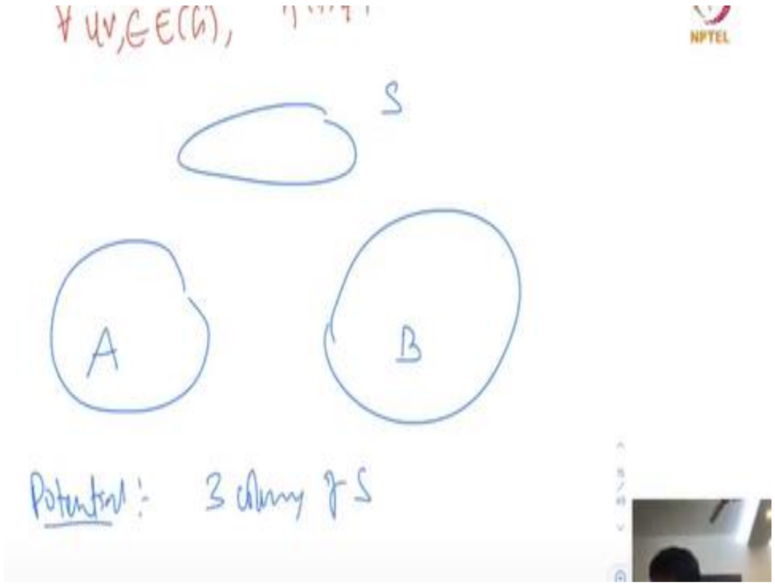
So, the next problem that I have selected is a 3-colouring problem. So, what is a 3-colouring problem, so before we even get here.

(Refer Slide Time: 01:24)



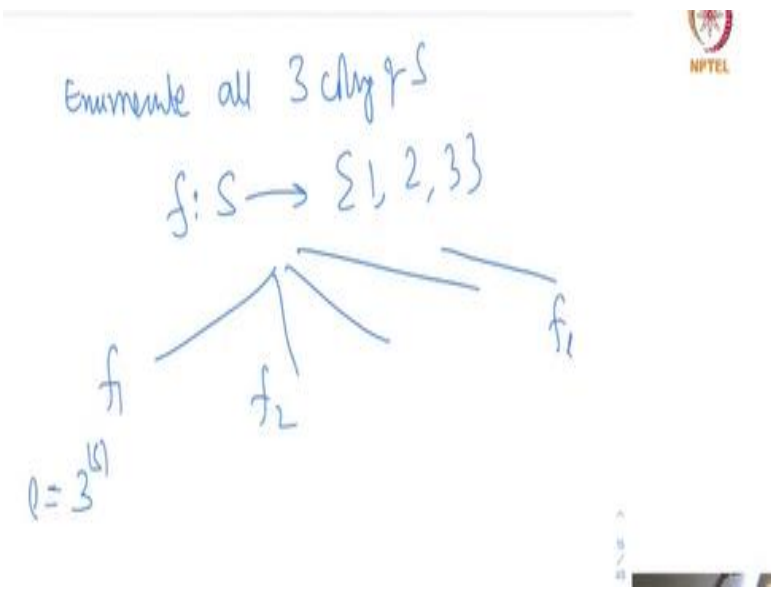
Let us try to understand what are we trying to understand. So, we want to solve 3 colouring problem. So, what are we looking for in 3 colouring, so you are given a graph V G you are looking for a function f from vertex set of 0, 1, 2 such that for every edge uv in E of G. What the property? f of u is not equal to f of v. So, I want to test whether the graph is three colourable or not on graphs of boundaries.

(Refer Slide Time: 02:02)



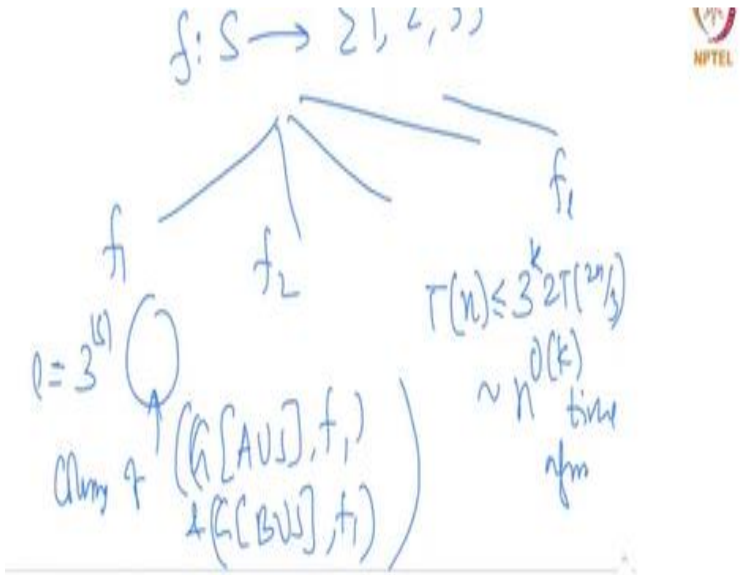
So, let us get try to see what kind of divide and conquer algorithm we could have gotten. So, suppose we had a separator S and these are the partition of my connected components into two pieces A and B, then we ask ourselves what are the possible partial solutions of S could be? So, we know that any partial solutions of S could be a potential 3 colouring of S.

(Refer Slide Time: 02:23)



So, the way we go about is that we enumerate all 3 colourings of S. So, you enumerate basically all function from S to 1, 0, 1, 2 or maybe it does not matter let us just put it just let us see what are we using 1, 2, 3 in the later so let us just say 1, 2, 3. So, let me change here also that not 0, 1, 2 but rather 1 2 and 3. So, we enumerate so, now you are enumerating such functions like f 1 f 2 f l where l is upper bounded by 3 to the power mod S. Now what are we doing after this? Once we have decided colouring here.

(Refer Slide Time: 03:30)



What are we looking for? So, we are looking to find a colouring we are looking for in this what are we looking here we want to find a colouring of two classes of graph. Basically, graph induced on say A union S and graph induced on B union S and a graph G has a three colouring if and only if graph induced on A union S and graph induced in B union S has a 3 colouring. So, it will take f 1 and take f.

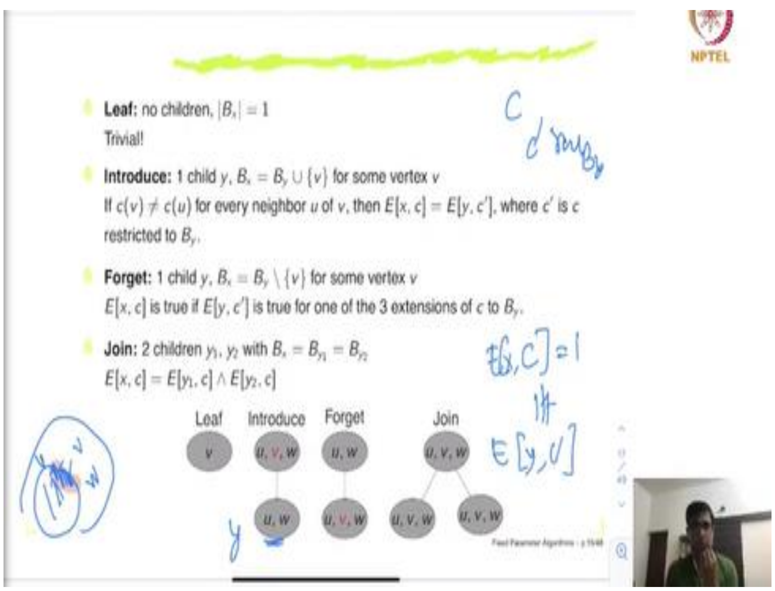
So, you are looking you are basically looking can we extend the 3 colouring of graph induced on S given by f 1 to a 3 colouring of A and similarly for B. Now so basically this is kind of a partial extension of this 3 colouring and if f 1 can be extended to a 3 colouring of this graph and if f 1 could be extended to 3 colouring of the other part. Then we know that there is a 3 colouring of the whole graph.

So, basically if you do recurrences then this will lead to T of n is like some 3 power k + 1 if the size of f is k and at most 2 times T1 by 3. So, this will be again give you some n to the power big of k time algorithm. But this tells us that what are we looking for as a partial solution at any point of time. That is basically three partial 3 colouring of my graph. So, with that in our mind, so we will see how to do this.

So, for B x is a vertical appearing in node x as before and capital V x denotes a vertices appearing the subtree rooted at x and for every node x and colouring of B x to 1, 2, 3 we compute the following Boolean variable which is true if and only if c can be extended to proper 3 colouring of graph induced and V of x. So, we start with the 3 colouring, so this is a node here and c is a partial 3 colouring.

And we are asking where, what are this filling is? This is one if there exists a extension of 3 colouring meaning you can come up with another colour which is identical at f. So, this is a next colour and that colouring has a property that for any edge. If you look at the end points of colours of the end point then they are distinct. So, now the question is now you are given the values for the children, how will you determine the E x, c for a particular bag or particular node.

(Refer Slide Time: 06:29)



And that is very easy because if there is no children you are leave then what are they going to do? You are going to do very simple is that you look at a particular colouring if it is 0 like so the particular colouring is going to assign either 1, 2 or 3 and that is trivial. Now look at any introduced node, so what is introduce node? So, what is node as always, I have told you introduce node means that there is a bag u, w and a vertex v is introduced.

So, you are getting a new graph with edges of v being either restricted to either between v and w. So, this is how the world looks like. So, now I have assigned v so given a child y which is this child y and, so what is this? We say look fine whatever may be the partial 3 colouring of this B. Well, it has to be a 3 colouring, so you start with the partial 3 colouring. So, it is like if I forget the colour of v then that provides a partial 3 colouring of the bag y.

So, now I ask myself if I assign this colour I know and where are its neighbours where are the edges its edges are either to u or to w or to none. So, if I ask is this colouring extendable, so this colouring is extendable if and only if what you call c, B is not equal to c, u for every neighbour u of v. So, I have a colouring c given. I look at the colouring c prime which is restricted to here the vertices contained in this bag in particular u, w.

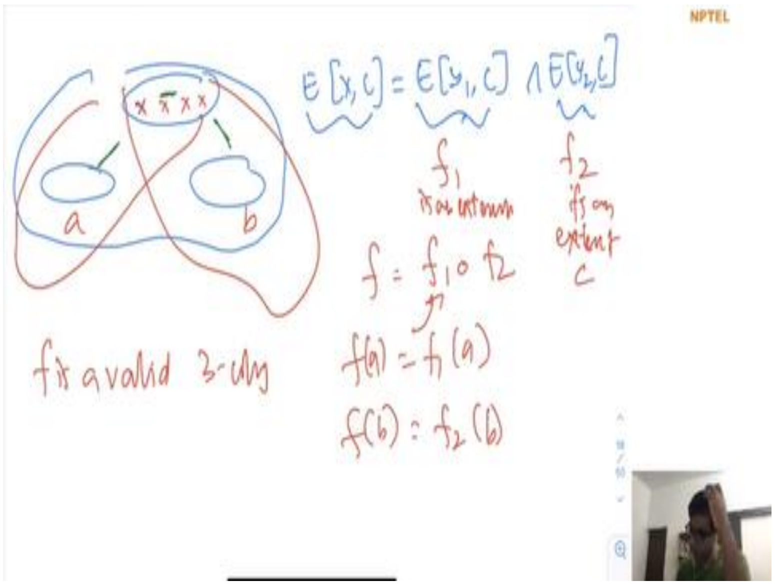
And I say is that associates to one. Then I put it 1 now whatever forget child. So, forget child if you are forgetting a vertex v. So, I am going to assign a colouring I say look at all the possible extension of this colouring. Meaning if I have a colouring c given to me then I say look E of x, c is true is 1. If and only if what is 1? E y, c prime is true for one of the three extensions of c to B y.

Because I know that look if it is a 3 colouring valid 3 colouring then the child which I am forgetting. Also gets some colour either 1, 2 and 3. And then what is c prime? c prime is the extension you fix c and for the vertices which you are forgetting say view you assign O. Is there a colouring where the v gets 1? If that is answer is 1 then I put no problem if this. What happens if it gets v? Gets 2 is that evaluated to 1 so on and so forth.

And now whatever join node is what does the join node means? Basically, join node is like this. So, I have a 3-colouring extension to the whole graph. If I look at this graph induced on this and this graph induced on this which is basically, I look at my child node with the same colouring c y 1, c said, does that evaluate to true? Well yes if that evaluates true that already is stored here is the other child with the same c evaluates true.

If it evaluates to true then I know that I evaluate meaning I have a 3 colouring for the whole graph. So, 3 colouring of whole graph, if and only if 3 colouring of the left side given here 3 colouring of the right side given here stored. So, if both of them are one then I know there gives a 3 colouring for the join and that is it. And we can actually formally prove some of these things.

(Refer Slide Time: 10:39)



So, let us just try to show this. So, it will help us so, basically what is this this is a join node. So, now how do I so? Look now notice what is the meaning that this evaluates to and this evaluates to one. What is the meaning of this? Meaning of this is, what does there exist a colouring of graph below this with colouring thing. So, some colouring has been assigned to these guys, I asked myself. So, now what can if E y 1 of c evaluates to 1.

It means there exists some colouring of the graph induced on this. So, let us call that colouring f 1 let us look at the and; what if this evaluates to? It means there is a function f 2 which evaluates to true. So, now I will show that if like this is and so if both of them evaluates to 1 then this definitely evaluates to 1 and if this evaluates 1 then we know there just a colouring which evaluates to both of these will able.

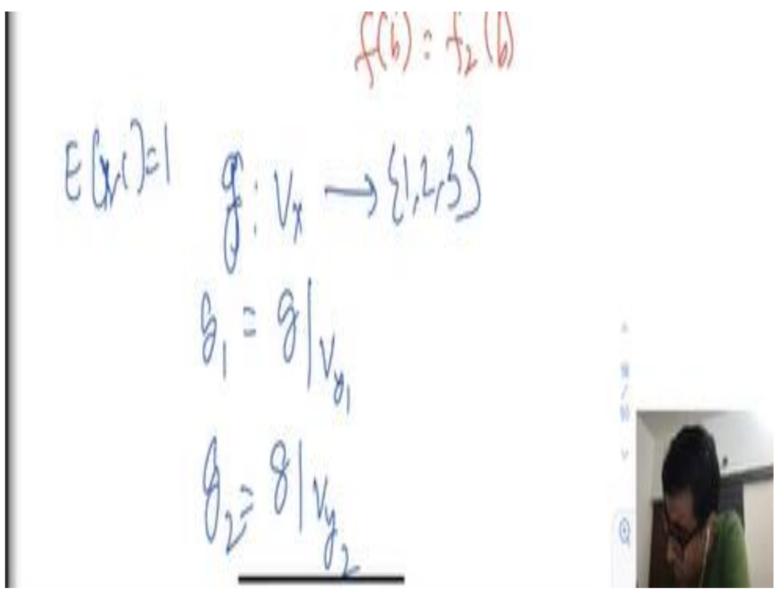
So, suppose if this evaluates one means there is a colouring f 1 which has this property. Then the colouring f 2 which has which are the property which exit. Now I will say just take a new f which is basically f 1 O f 2. This is basically concatenation let us say concatenation. So, basically what is that? If for any vertex y which is not here you follow f 1 for y let us say. So, suppose I had a vertex a here and some vertex b here in the this side of the graph.

So, what is f of a is nothing but f 1 of a right if b is this side what is f of b is nothing but f 2 of b. And if on the common vertices both f 1 and f 2 have the same colouring. Because f 1 is an extension of c and f 2 is an extension of c. So, both f evaluates so now all I have to show that f is a valid 3 colouring. So, if you give me a valid 3 colouring of this portion this portion then I can get a valid 3 colouring for the whole graph.

Now why this is look at any edge so either an edge, where are my edges? My edges are contained here or edges are contained in the left-hand graph or the edges are contained inside the right hand graph. Because now every edge is present in one of the graph and so if this edge is present here then f 1 helps us f 2 is present here then f 2 helps us because look at those endpoints. If f does not assign the correct colour.

Then look at the map in f 2 but that should also be valid violated because we are just following f 2 but that does not happen. So, given a 3 colouring for the left hand side and given a valid 3 colouring of the right hand side we can get a valid 3 colouring for the whole curve. Now let us see what happens.

(Refer Slide Time: 14:50)

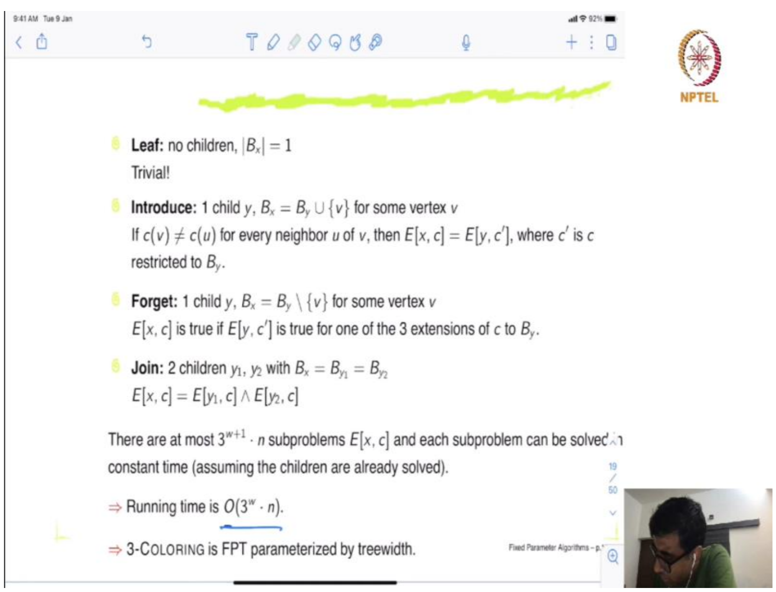


So, suppose E x, c is 1. Now look at the colouring suppose this is 1 it means there is a colouring f from let us say v x to 1, 2, 3 which is valid colouring. Now look at the colouring I will say let us say let us call it g 1 look at a g 1 colouring which is restricted to this side. So, g 1 is like g 1 is suppose this this was like this is y and this is like y 1 y 2. So, graph g 1 restricted to g 1 is nothing but g restricted to v y 1 and what is g 2 g restricted to v y 2.

So, what is the property of g 1? g 1 matches with v x, only left side g 2, what the property g 2 is matches with g 2 in the right hand side. But then what does this chosen? I have been able to show that there exists a colouring in y 1 which is a valid 3 colouring. It means definitely E y 1, g 1 E y 1, c because of existence of g 1 will evaluate to 1. Similarly, E y 2, c will evaluate to 1 because of existence of g 2.

So, this is how will you will have to show that all the recurrences are correct. So, given a valid solution to here I should be able to get a valid solution to my children. Given a valid solution to my children I should be able to get a valid solution to my left hand side. So, this is how so you should try to prove it formally all these things and get.

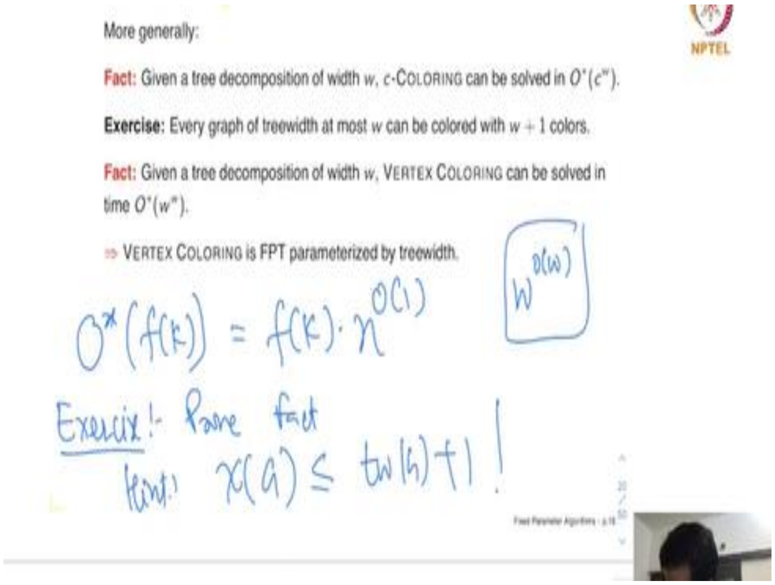
(Refer Slide Time: 17:06)



So, how many sub problems are we creating? We are creating 3 to the power w + 1 times n sub problem and each sub problem can be solved in constant time assuming the children are solved.

So, you can actually get an algorithm with running time O 3 to the power n. And that so the three colouring is FPT parameterized by treewidth.

(Refer Slide Time: 17:32)

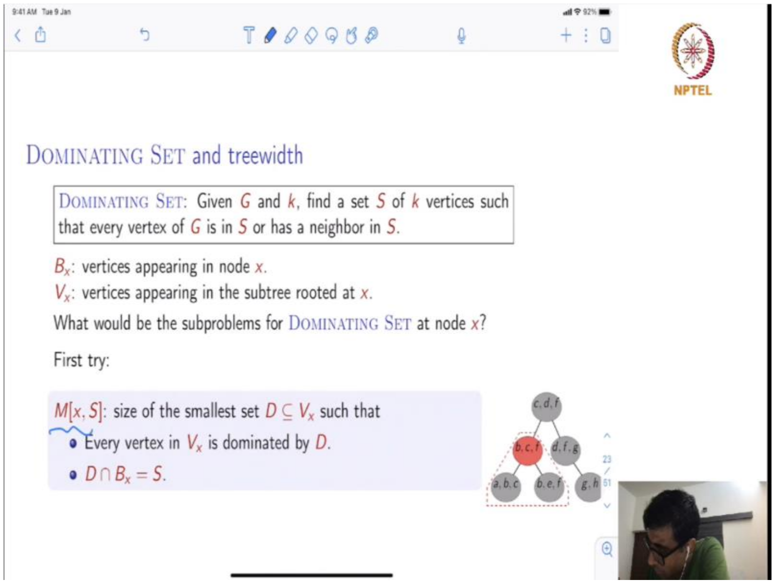


So, what about more generally given a tree decomposition of width w you can do the same way c colouring in c to the power w. And basically, this O star notation is used in parameterized complexity to O star notation is used. So, when we write O star sum f of k, then this basically is a shorthand notation for f of k n to the power b. So, we actually do not explicitly write the polynomial dependence on the input right.

So, now given a tree decomposition of width w vertex colouring can be solved in time w to the power w it is a good exercise. Prove fact and hint is so that the chromatic number of G is function of tree width of G and then you should be done. So, the other let us say so this is it. So, using this you should be able to prove it. But what is interesting is that this algorithm w to the power w is optimal.

So, hoping that we can get an algorithm with w to the power lit low of w is unfortunately not possible and hopefully we will have time to so such a reduction in the course.

(Refer Slide Time: 19:10)

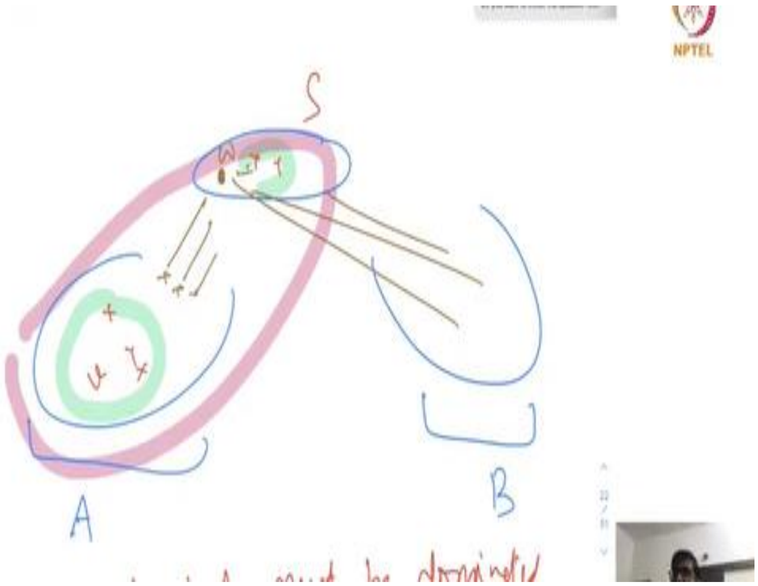


So, now we have seen two examples in the morning we saw max independent set. We saw example of max independent set in the morning weighted max independent set. Now we saw 3 colouring, now let us see another problem of our interest that is dominating set. So, what is the dominating set? That you are given a graph G and k you have to find set of k vertices so that every vertex of G is in S or has a neighbour in S.

As before B x will denote vertices appearing in node x and V x will denote vertices appearing the subtree rooted at x. So, what could be the sub problems for dominating set at node x. So, let us make a first try. So, whatever should be the first try? What is M x, S? What is an M x, S, is like I am looking at the node x and f is the subset of V x what we want to find a D subset of V x is that every vertex in V x is dominated by D and D intersection B x is equal to S.

So, I decide which vertices from this bag is going to be the dominating set I say look, is there an extension like what is the minimum size dominating set extending this S and that dominates everything in V x.

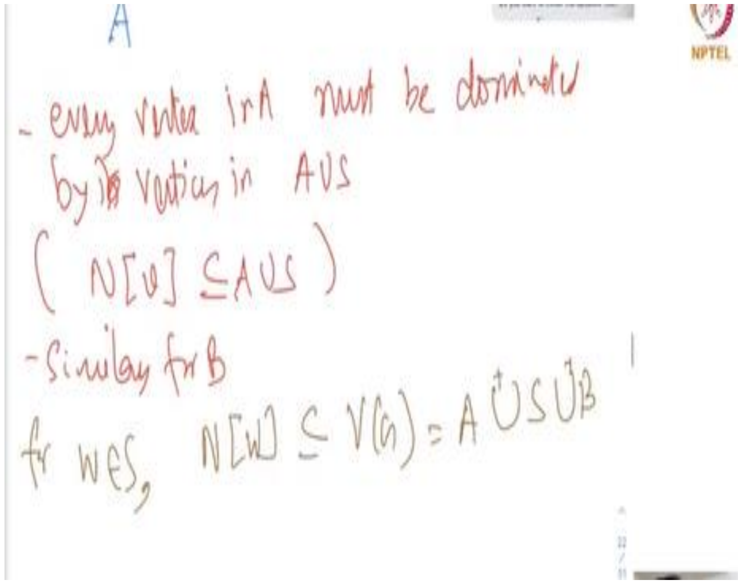
(Refer Slide Time: 20:55)



So, let us see what is your problem with such a try and because notice what happens as always, I have told you try to do divide and conquer. So, if you do divide and conquer then I have a separator and I have some vertices here and I have some vertices here. Now like this is like part A and part like this is a portion. Now look at a vertex here, where are its neighbours? Its neighbours are here or its neighbours are in my separator.

So, the only way we can get dominated is by if this portion. So, every vertex which is in A so what information we can get, every vertex in A must be dominated by its vertices in A union S. Why? Because N of v is contained inside A union S. And similarly for B, but now ask ourselves the following question, what about a vertex in here? Now this vertex call it w it could have a neighbour in a sum in S and some maybe in B.

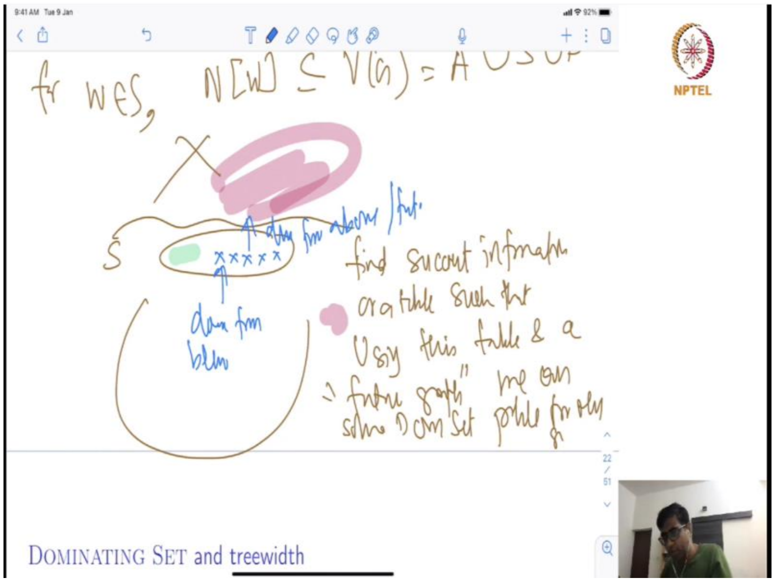
(Refer Slide Time: 22:45)



So, for w in S neighbourhood of w is contained inside whole of V G. Now or rather say it could be in A it could be in S it could be in B which implies that if I am solving recursively if I am trying to solve here, I may not like I may not need to dominate a vertex in S. Some vertices in f may be dominated from the B side. So, basically, we have to guess three set of information, which vertex in S is going to be in the dominating set that is the first thing.

And those vertices which are not going in the dominating set are they being dominated from A vertex or are they dominated from the B vertex or they are dominated from a S vert. So, you have to decide, so basically if I would have been doing dynamic programming algorithm then as I have been telling you the history.

(Refer Slide Time: 24:03)



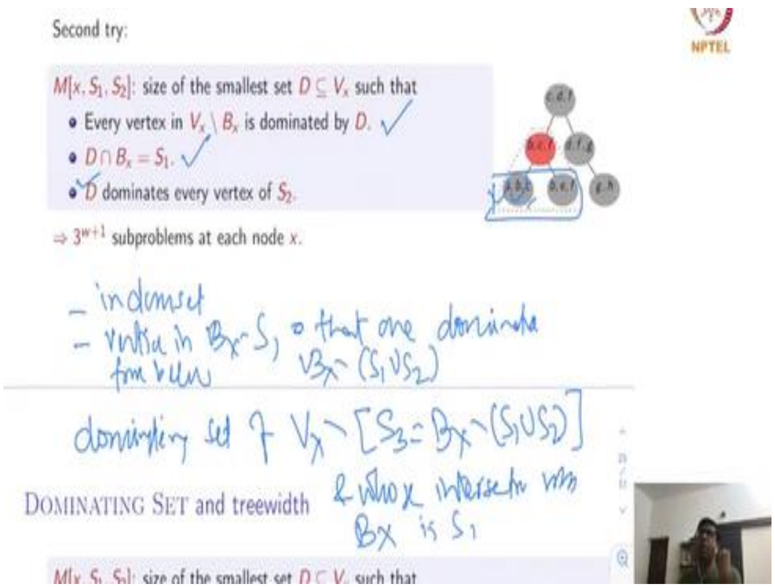
So, this is my bag and this is what it is. So, we need to find succinct information or a table such that using this table and a future graph. We can solve DOM set problem for whole graph. So, I do not see this portion, up so based on this information I only see when I am doing bottom of dynamic programming, I do not see future based on locally. I say I do not know for any graph, how this information look like?

But I am going to have sufficient information from the below, so that you give me any graph from the top. That graph will could just look at the look at the table he does not need to lick the below graph. And with the help of this table and the graph itself this graph it is the top graph itself it will be able to compute the minimum size dominating set of the whole graph. So, now I say look irrespective of whichever graph it is.

Going to intersect this S irrespective of top graph look at an optimum solution of any like of a graph of my current graph or that matter any other graph. It is going to intersect S what are the possibilities it could intersect it could take some portion here. And for the remaining vertices we have to decide dominate from below, dominate from ever future whatever. Then once you have this then you are basically looking for a dominating set for not every not the; whole below thing.

But you are looking for a dominating set of vertices which for all vertices but for those which will be dominated in the future. So, now we will try to make this intuition this into a table. So, that is our so as I explained to you problem is that vertices in V x can be dominated by vertices outside V x.

(Refer Slide Time: 27:02)



So, this leads to the following sub problem, so, we have at this node the partition S 1 S 2. So, the size of small is set from below so that every vertex in V x minus B x is dominated. So, every vertex here is definitely dominated by D. D intersection B x is equal to S and D dominates every vertex of S 2. So, you get my point, so we have two things S 1 and S 2 and what are they like so as I told you some vertices here could just be dominated by these green vertices.

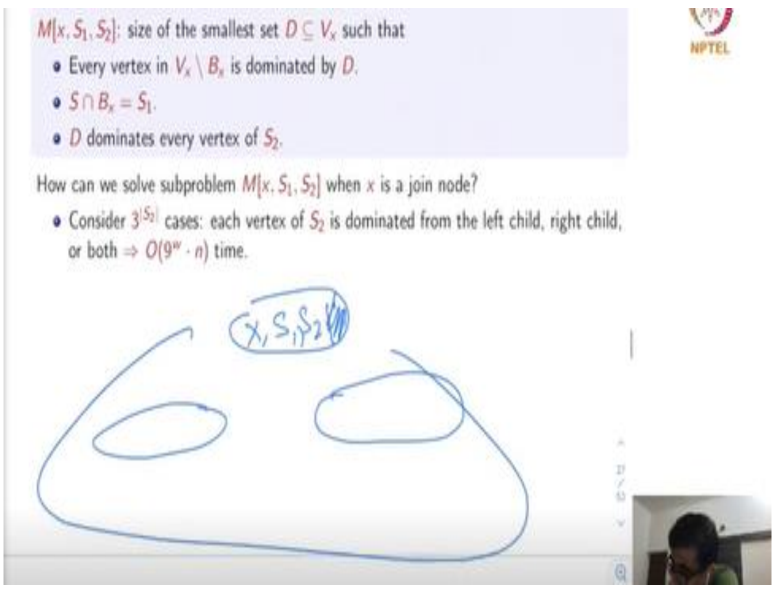
So, dominated from below which also mean that they are probably dominated from here. So, this is what we are trying to write here. It is a size of the smallest set contained from below which has some property every vertex from here is dominated by D that is one thing. D intersection you are going to pick S 1 from here and D dominates every vertex of S 2. So, how many such problems are you going to create and the vertices.

So, what is this you are going to be so for every vertex what are the three states? For every vertex that states R in dominating set that is D intersection B x is dominating set. Now so, those are automatically vertices in the B x minus S 1 which is like, what is that are dominated from below. So, they are not part of the dominating set but they are dominated from vertices in the V x. So, in particular they are dominated by D.

And what we have not care about dominating the vertices in B x minus S 1 union S 2 are those vertices in this picture which will be dominated from future. So, there are some vertices which will be dominated from outside. So, you are basically in some sense you are trying to find a dominating set. So, if I have to write in other alternatively, what are we trying to do trying to find dominating set of V x minus I would say S 3 which is nothing but.

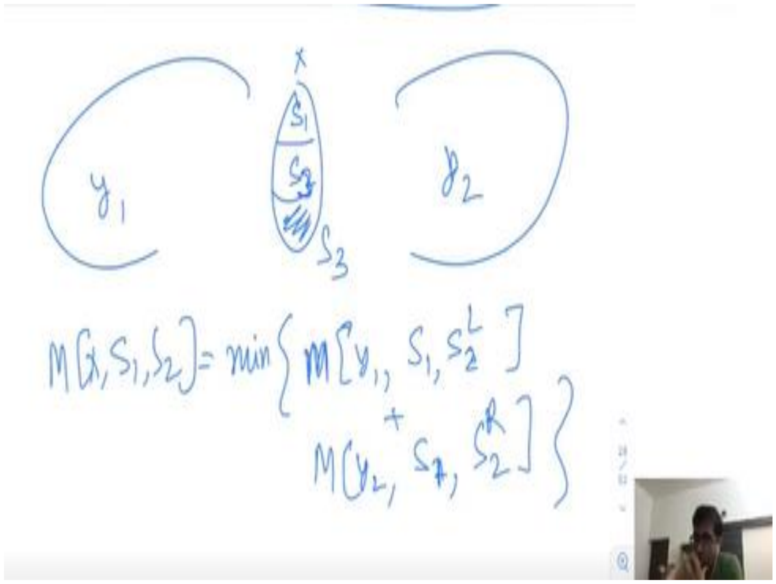
So, you are dominating everything but this and whose intersection with B of bag of x is just one. So, look at the vertices in the bag some of them goes to this dominating set. Some they do not go to the dominating set that is fine. But among those I decide which one of them are dominated from below and which of them will be dominated in future or will be dominated by vertices which is not present in V x. So, once you made decision then you are perfectly fine.

(Refer Slide Time: 31:17)



Now so the question is how can we solve sub problems M x, S 1, S 2 when x is a joint node? Rest of them are very similar. So, what you know? Each vertex of S 2 is dominated from the left child, so look at join node here it is. So, remember in join node, what is your goal? Now you have x, S 1, S 2 and the sum people you do not care. But now you have to decide, what is the property? S 1 is the set of vertices which is going to be the dominating set. Now you decide for S 2, I do not know about S 2.

(Refer Slide Time: 32:23)



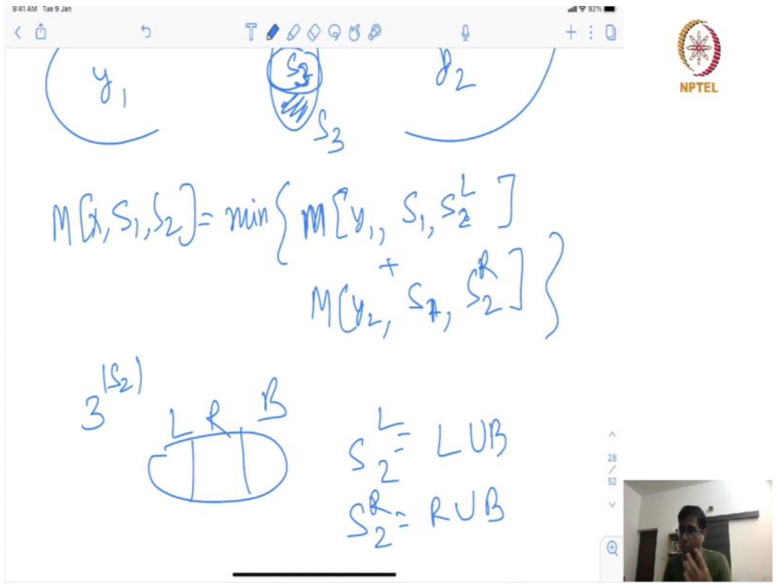
So, I want to fill let us try to fill M x, S 1, S 2 I have to decide. A vertex in S 2; look remember what is your criteria? Vertex in S 2 needs to be dominated from somewhere because we are looking for a solution for the whole graph. So, if you look at a graph perspective look at this graph perspective, so the way actually the way we make is this here it is. Even better let us draw like this and this. So, now this is my bag this is my node x.

This is my portion of S 1 this is my portion of S 2 and we do not care about this. So, basically it is a S 1 is you are in the dominating set you are dominating from the left S 3 is something which I do not care whether to dominate. Now look at S 2, now I do not know from S 2 so if I have to fill S 2, so I ask myself look M suppose this is of child y 1, y 2. So, M y 1 and so what is this going to be minimum of M y 1, S 1.

Now I decide for S 3 maybe some of these vertices should be dominated from the left side some of these vertices may be dominated from the right side. So, now what I will do M y 1, S 1, S 2 say left plus M y 2, S 1, S 2 or maybe both I do not care I mean maybe you are dominated from both. So, basically you are you have to go through all such possibilities and take minimum, so basically how many such sub problems will you generate.

You will generate as many sub problem as many as partition of like division of S 3. And that is why you will generate like S 2 maybe you can dominate it from the left or you dominate it from the right or dominated from both.

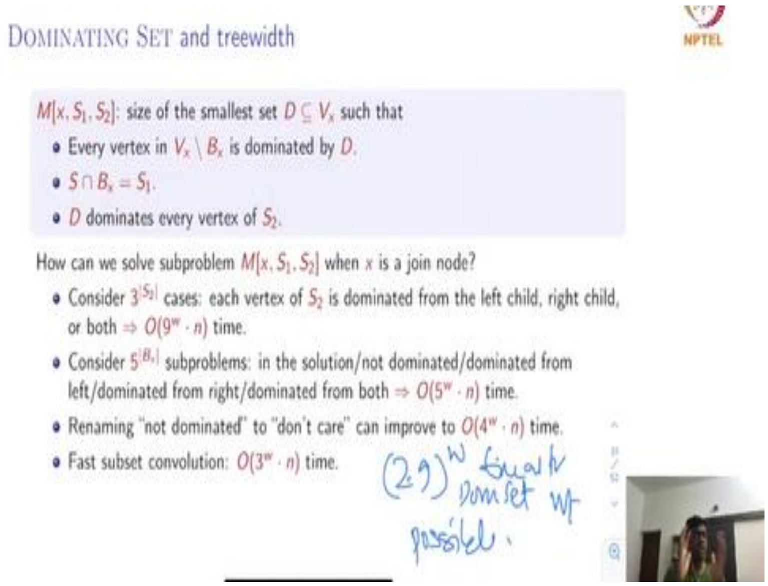
(Refer Slide Time: 35:07)



So, that will tell you three partition of S 2 and that three partition like from the both. So, if you decide for S 2 left right both. Then what will be the S 2 left is like l union both, what is our S 2 right is right union both so you put this. Now you say you compute these values for each of these values you compute the number and then you have to subtract the S 1 value cognitive S 1 because that is common to both and among all such possible values you will take the one which minimizes.

So, now notice to fill this table how much time will we take. We will take to fill each of this table we will take 3 to the power mod S 2 time. So, in worst case this could lead to like 9 to the power w and time algorithm to do this. And a trust me like introduce node forget node are much more easier than this.

(Refer Slide Time: 36:20)



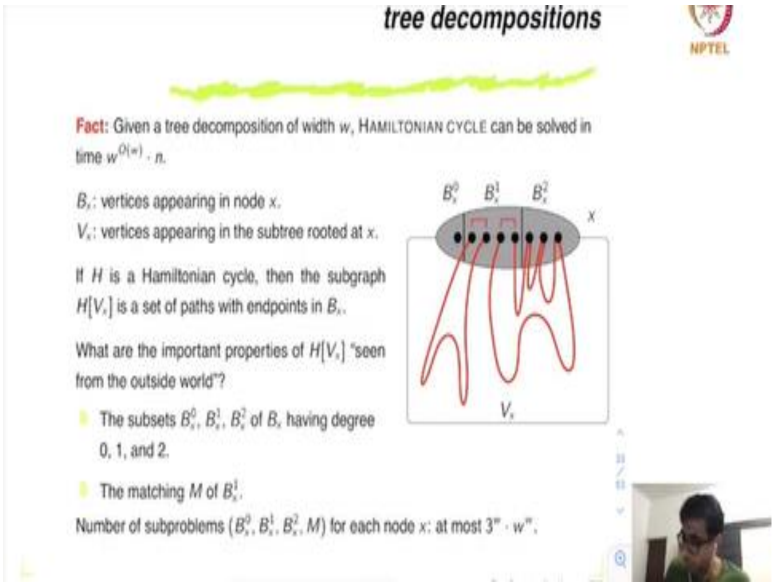
And so, this implies that if you do this it could lead to you know 9 to the power w kind of algorithm. There are ways to improve all this but like let us I will not tell you all this I mean you can get down to some 5 power w 4 to the power w I like by some clever simple tricks with who is dominated from which side or not. But let us not worry about those things in fact we can get down as good as 3 to the power w.

So, like this join node looks like that we create to fill one entry we are creating lots of sub problems which is true in some sense. But we can do some very clever way of filling these tables and actually we can get the running time of the algorithm for dominating set parameterized by tree to just 3 to the power w which is cannot be beaten. In the sense that we will be able to show later that you do not have say for example 2.9 to the power w time algorithm for dominating set for example.

Of course, under some conjecture not possible. So, but I just wanted to give you the feeling of you know like you have to have a feeling of notion of how partial solutions look like what is the notion of left right and all possible partial solutions basically creates an entry in the each possible partial solution a correspond to entry in the table. And the way we make a table is that given any right hand side graph or the given any future graph.

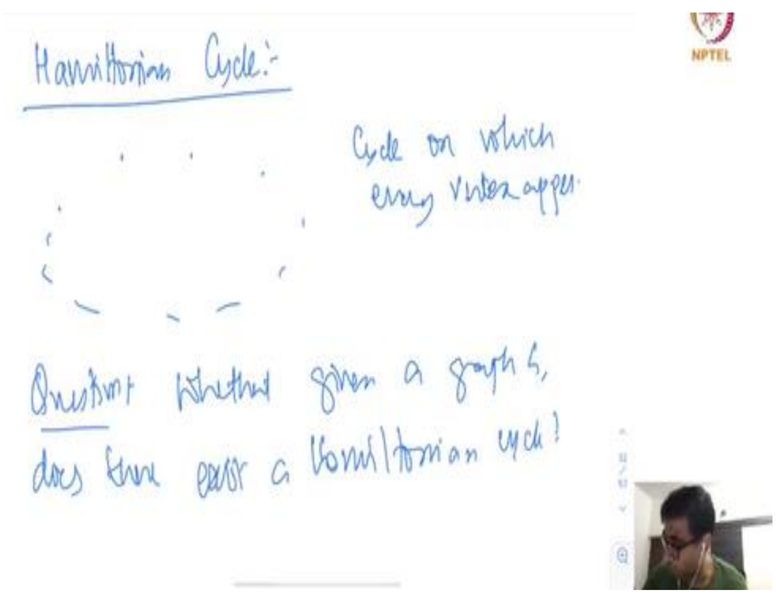
And this table it can compute the solution of whole object just by looking at this table and looking at the right-hand side.

(Refer Slide Time: 38:12)



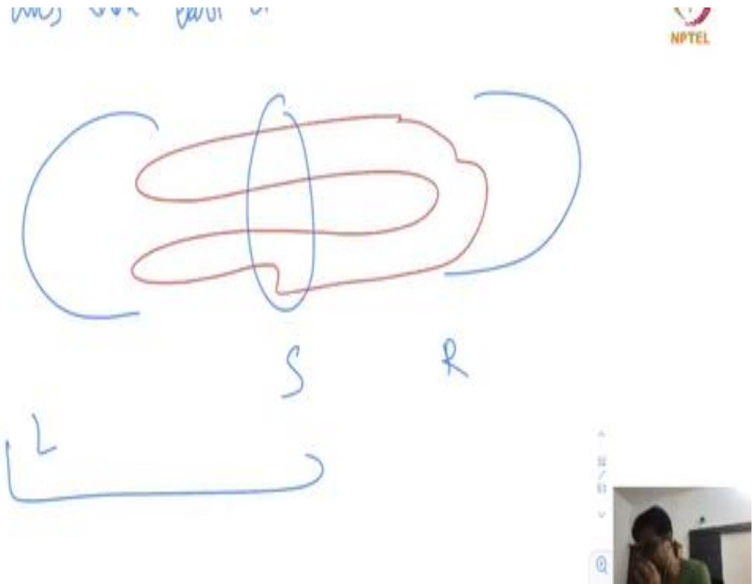
So, the last example that I want to talk about is Hamiltonian cycle. So, let us try as like as a thought process I have always told you that like.

(Refer slide Time: 38:33)



So, let us look at Hamiltonian cycle. So, what is Hamiltonian cycle, is basically a cycle on which every vertex appears. So, a cycle is called Hamiltonian if every vertex is present on this. And the question is whether given a graph G does there exist a Hamiltonian cycle. Let us do our thought process again.

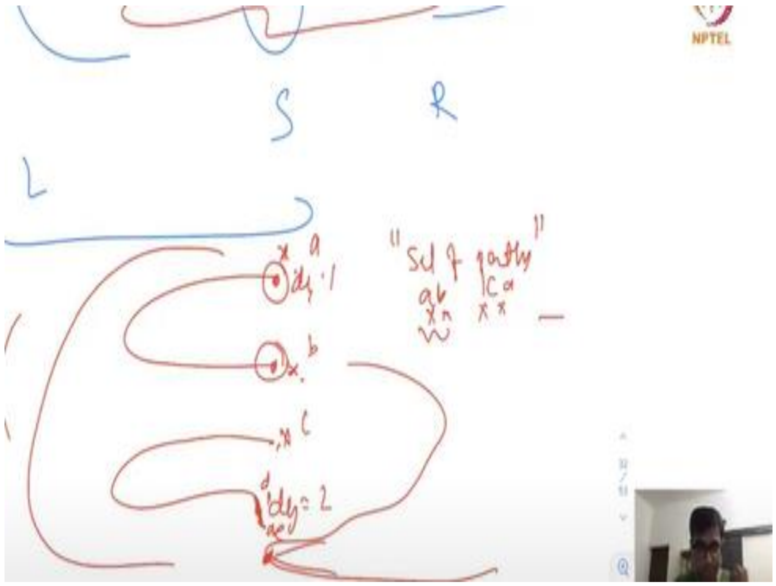
(Refer Slide Time: 39:22)



Again, our thought process is our separator left side right side. Now what is the notion of what is the meaning of partial solution? Partial solution basically means how will the how will if I look at any solution and if I look at the intersection here how will it behave. How will it look like? Now look at any Hamiltonian cycle how could a Hamiltonian cycle look like a Hamiltonian cycle suppose look it touches every vertex. Maybe it comes like this like this like these passes maybe.

I do not want to draw such nice picture, it could be just look like this. Maybe just to make our life little bit easier let me try this is slightly more nicer to have a look. Now notice if I look from the left hand side perspective what do I see.

(Refer Slide Time: 41:00)



I will see something going and again going after this. Now notice that once you have drawn this, so this is what we see. So, what is the property? The property is that some vertices says look at every vertex. So, we ask ourselves if I look at the left-hand side what are we looking for what is my partial solution. So, partial solution is basically some set of paths and what is the property of these paths prop these paths somehow like start in the root like in the separator bag.

And end in separator bag in some sense and so, what the property that these vertices if I look from the left side they are getting one edge on the simultaneous cycle from the left side. So, its degree is 1 but some vertices for example here look at this it is a separated sign but what is its degree is 2 because it is here. Maybe there is some vertices here whose degree will be 0 because maybe they are getting like the cycle is coming in some way and they are going completely.

So, we are looking for an Hamiltonian cycle but will have some property, we will specify you. Look these are two degree one vertices it means you give me a cycle from below cycle like a set of disjoined paths with a property which spans every vertex from here, if its degree is non-zero. And it has this property that it starts from here it ends here. So, I will also tell you the set of parts what are its end points in this is like I will tell you it starts at this.

So, it is like I will give you pairs. And I will look I will tell you how can I define these things I say look at these pairs, so for example here a b is one pair c d is maybe another pair or something or maybe d is another pair. And for these things we will some other specification and we will say is there just a set of disjoined paths joining a to b c to d and maybe d to e or whatever may be when you call it this and for the vertices on the bag which is a zero, I will not care.

I am looking for this disjoint path that covers every vertex in the left side and as this property. So, this is how a partial solution will look like and given this like any partial solution can be described so it can be described by defining some set of pairs defining some degree 0 vertices degree 1 vertices and so on and so forth and trying to find a set of destroying paths satisfying these specified properties. So, this is what is being done when we try to do this.

So, if you look at this now, we are looking from this perspective of vertices appearing in node x. V x vertices appearing in the subtree rooted at x and if H is the Hamiltonian cycle, then the sub graph H V x is a set of n set of paths with end points in the back containing this. So, now notice what is B x 0 these are those like when I look at only edges which is contained inside this then the edges here are not there. So, from the down perspective these vertices are of degree zero side.

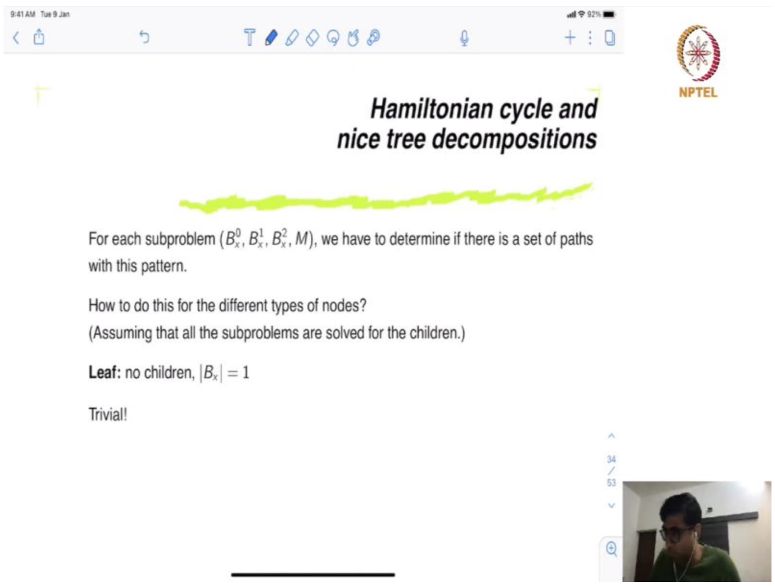
So, from the outside world because the Hamiltonian cycle is going to going to pick up like this. So, those are degree zero vertices different. Now what is the property of degree one vertices? Here vertices is basically these are like pair. So, from the it goes and the cycle leaves this and completely crosses immediately to the other side. So, you will have these vertices will have degree one, so these will form a pairing, this will form a pairing.

So, what are the vertices of degree one are those vertices which uses one edge this side and one is on the top. So, and some of these vertices here are degree two this basically I mean they are degree two in the sense it means that they are both the end points are in the left are in the below side. So, this is what we are trying to look for. So, the subsets B x B x 1 of B have degree 0 and 1, 2 respectability and we have a very specified thing that there is a matching on B x 1.

But so, degree 1 vertices should be matching because I am coming in from one and I am leaving from the other similarly. So, this is the pairs which you will get, so for example here a and b and maybe I did not draw it properly otherwise c and d would have been the corresponding pair. So, how many such sub problems are there, so basically first you partition them into degree 0, 1 and 2 so that is like 3 to the power w.

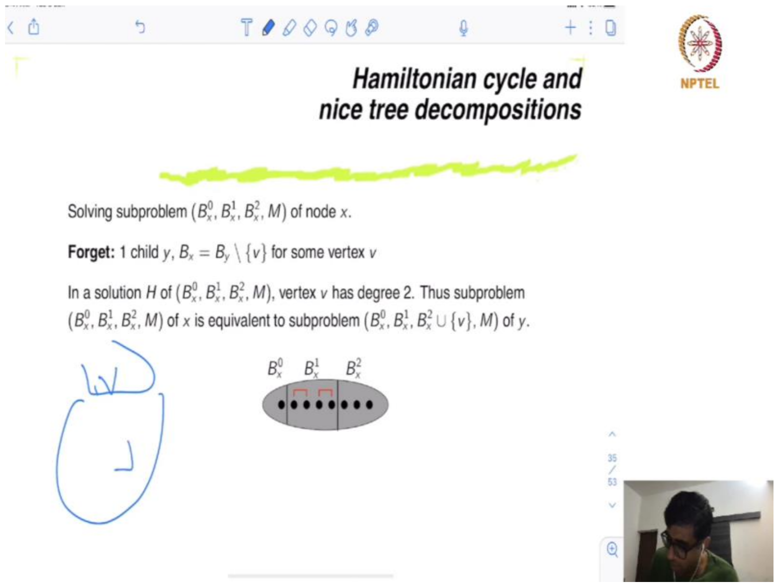
And now for the degree 1 vertices you have to find the pair. So, basically it is a partition of some subset of vertices into pairs, and that is upper bounded by w to the power w such pairs. So, you are specified by three set of information which vertices are degree zero which vertices is at a degree one which what is say the degree two and what is the pairs of what is the pairing in the vertices of degree one.

(Refer Slide Time: 46:54)



So, for each sub problem we have to determine if there is a set of paths with this pattern. How will do this for different types of nodes assuming that all the sub problems are solved for its children. So, again you will see what can we do in introduced node what can be doing forget node so on and so forth. So, no children B x equal to one trivial it is a single vertex so you can fill up everything.

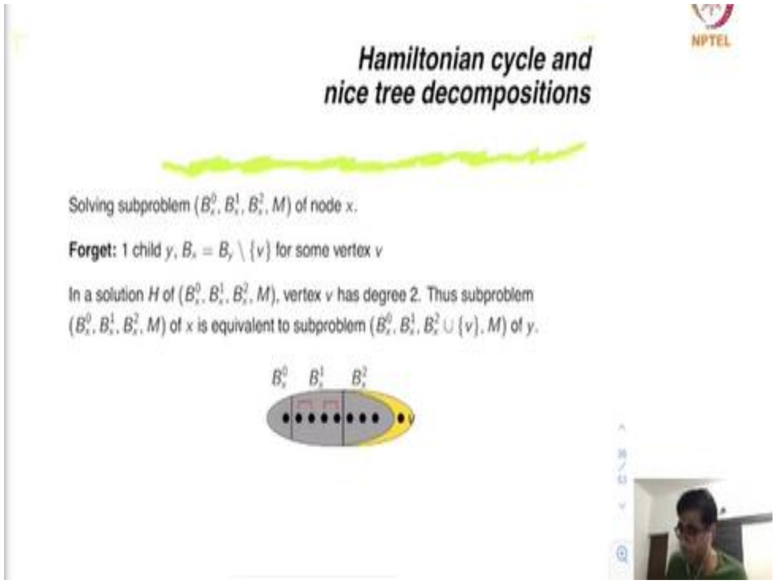
(Refer Slide Time: 47:17)



Now look at a forget node, this is slightly easier so what is happening. So, if you are forgetting for some vertex v then what is the point if I am forgetting a vertex v at this point of time which basically means that after the all the like v does not have any more neighbour outside. So, v has all the neighbours inside, which implies that if I am looking for some characteristic vector here then I should put v equal to we put partition v into degree 2 and check like.

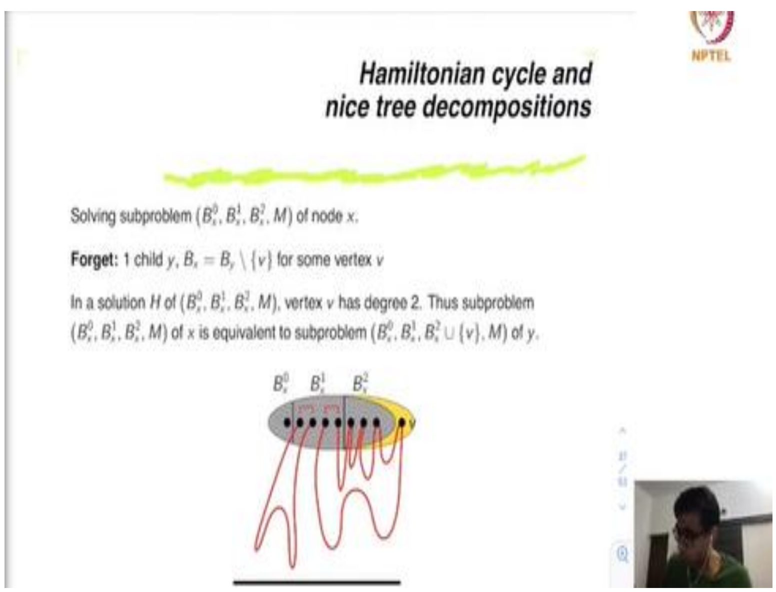
So, the sub problem it will be like okay fine v is missing like this is a forget node I just put v in degree 2 that is it and with the same pattern. So, this will exist if and only if this. So, for a node below we have already computed and we check if this is 1 or not, if this is 1, we put it 1 here. That is, it,

(Refer Slide Time: 48:17)



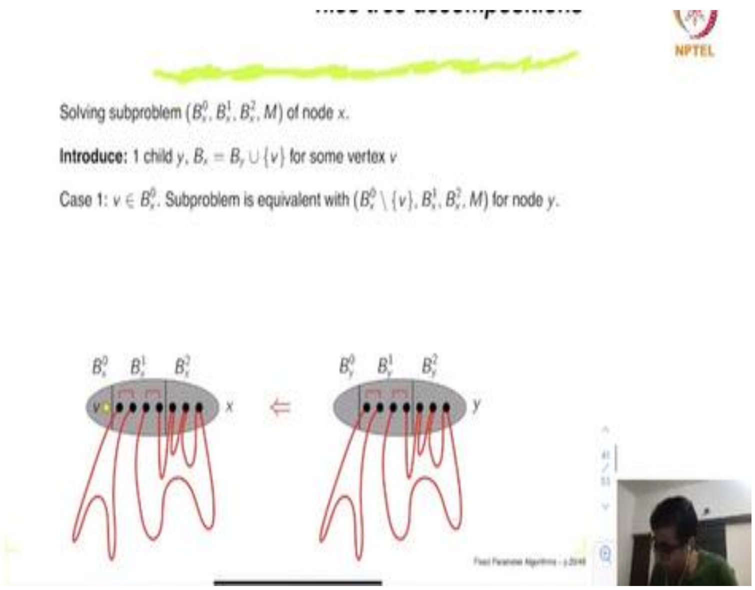
So, we put v in the degree 2 and ask the sub problem is equal to the sub problem same degree zero same degree one. But now we are forgetting this v vertex it means the Hamiltonian cycle will contain both the edges incident to this v here itself. In other word I will say is there a disjoint path with this pattern where v has degree 2 which we have already stored in the below, so based on its answer we will put 0 over 1.

(Refer Slide Time: 48:47)



So, we are asking is there a pattern and if there is a pattern of this nature then we say one or we put zero.

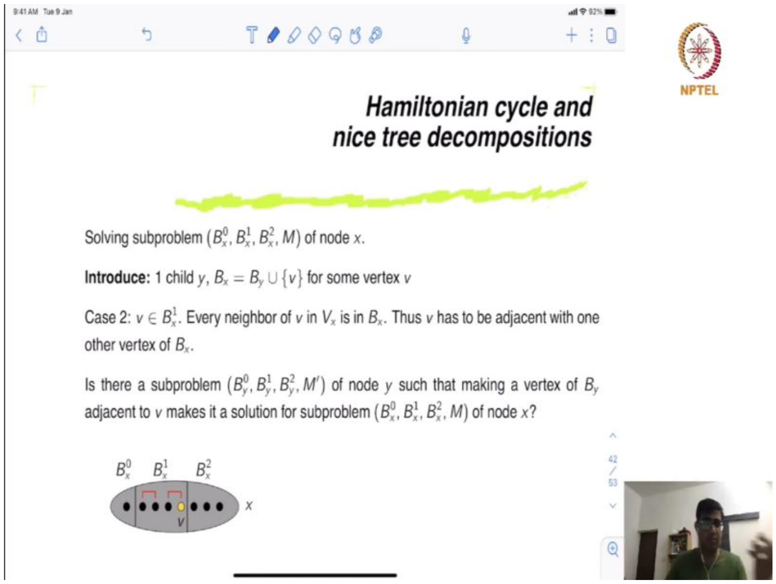
(Refer Slide Time: 48:59)



So, now let us look at the introduced node. So, introduce node will have several cases either the easy case is that B belongs to v has degree 0 then it is equivalent to that I mean if v has degree 0 then you are looking for a pattern without this vertex. Because that vertex is not appear this vertex B does not appear on the path system that we are looking from the below. So, you put minus v 0 B 1 X B 2 for node y and that is it.

So, if you forget this and you look at this if this is 1 then you put 1 otherwise you put zero. So, if we have such a sub problem for node y then you put then you are happy otherwise you are not. Because if you had such a solution that implies like you just take the same disjoint solution here like same pattern solution the same pattern here, that will also correspond to the pattern solution when v has been assigned zero. So, a solution here corresponds to solution here.

(Refer Slide Time: 50:12)

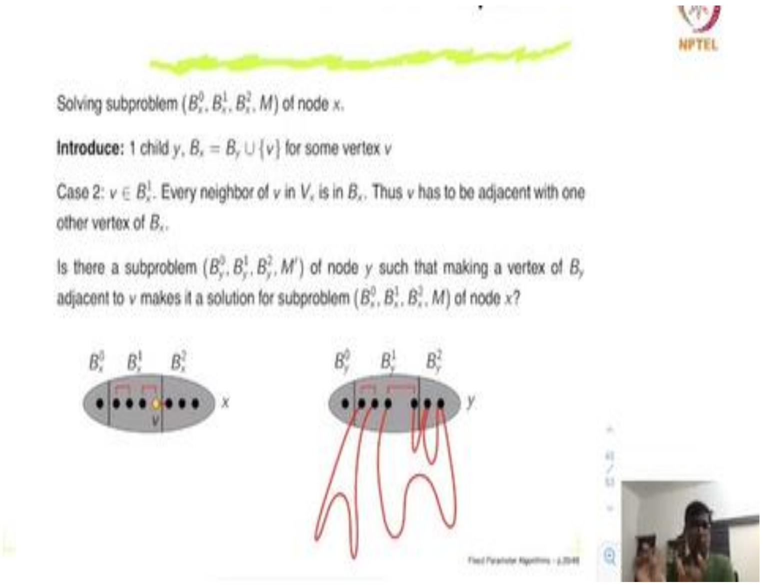


Case 2 now a v in B x 1, what is the property of this? We know that every neighbour of v in v x is in B x. Because this is the time when a vertex is introduced then all its neighbours at that point is in this bag. Thus, B has to be at least adjacent with one other vertex of like of B x. You get my point the point is that look what is now if I am going to ask v has degree 1, if v has degree 1 which implies that one of the vertexes its neighbour has to be adjacent with one other vertex of v.

So, let us ask is there a sub problem B y 0 B y 1 B 2 y M prime of node y so that making a vertex of B y adjacent to v makes it a solution for a sub problem B x 0 B x 1 B 2 x M of node x, you just ask yourself. Then for different choices, now so this will correspond to look this is degree one, so the only possibility is that his other neighbour look I mean it could just be an edge here that is also fine so, but this v could so you are making a pattern here.

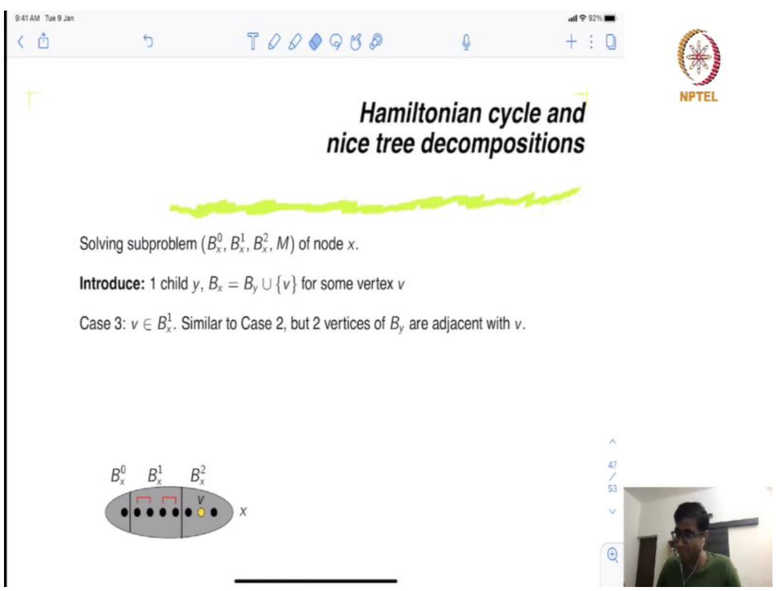
So, either this vertex will get its neighbour in this from B to x then which could look like this or you could have a pattern like this and this and it is getting one from here. You could also be that you know it could be just like this, so then it is just like the pattern itself is this.

(Refer Slide Time: 52:20)



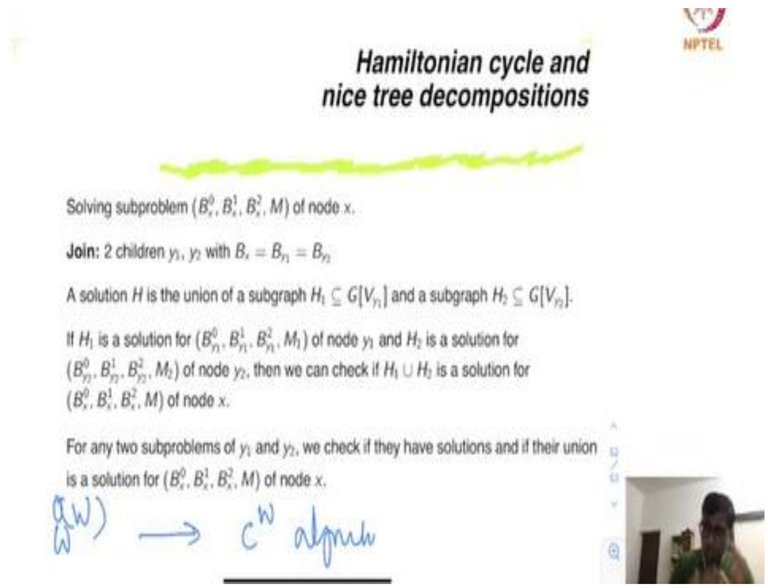
So, several cases will appear and you have to handle each of them cases by looking at what is a problem to make what is a problem not to make and so you do it case by case for everything and you deal with. I am not going to details have a look at these I mean these slides are properly done it should help you.

(Refer Slide Time: 52:42)



So, the bad case was when this was degree 1 and if v is in B 1 x but what is the property that if v is in B 1 x and then similar to case two but now two vertices of B y are adjacent with v. And again, you deal with them okay. So, there will be several cases when you introduce comes and you have to deal with each of these cases but like I will omit this you can have a i will provide these slides you could have a look at it.

(Refer Slide Time: 53:23)



And similarly, if you have two children y 1, y 2 with B x B y B 2 and then a solution H is a union of sub graph this and the solution this. So, you have to check whether their union is solution at node x, so for any two sub problems y 1 y 2 we check if they have a solution and if the union is a solution for this node x or not. And basically, H 1 will provide you some sort of connectivity H 2 will provide you some sort of connectivity.

And you see can you achieve this part of the solution using all these things. If you are able to succeed then you are happy if not then, so T s will probably also spend some time doing on this. But all I want to tell you that you could do all kinds of problems doing a simple dynamic programming algorithm. And if you do this the way we have done is like you will get w to the power w kind of algorithm here.

But after lot of hard work, it is possible to get some c power w algorithm. But that requires different techniques and different methods and that we will not talk about in this course. So, I hope I have given you some idea and some way to do dynamic programming algorithm over drafts about it, and I hope you can make use of it. Thank you.