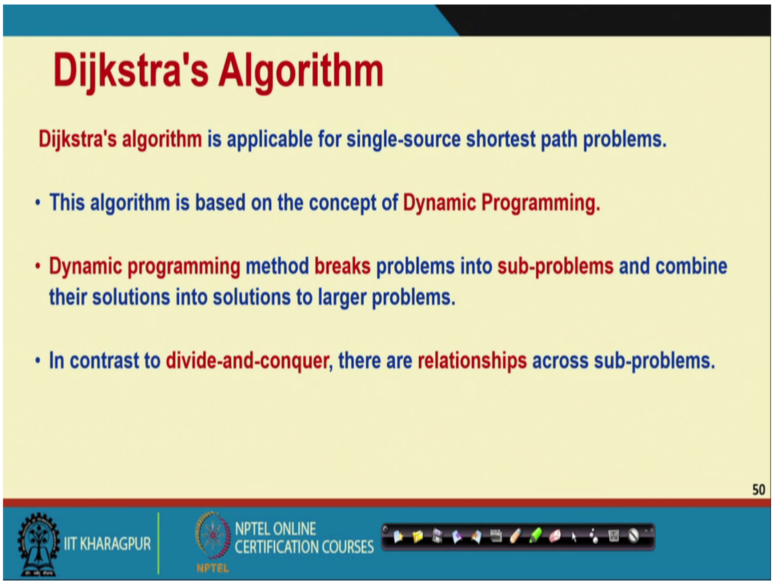
Selected Topics in Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture - 10
Dijkstra's Algorithm

So, a good morning to all of you in our course Selected Topics in Decision Modelling. Now we are in our last lecture as far as the dynamic programming is concerned and this topic is Dijkstra's algorithm. And Dijkstra's algorithm basically is a network application, application in the graph or networks. This is really another; what you call algorithm for obtaining shortest path. We have already seen some shortest path algorithms through dynamic programming. Now, you may question that why one more you know algorithm, we are again going to study.

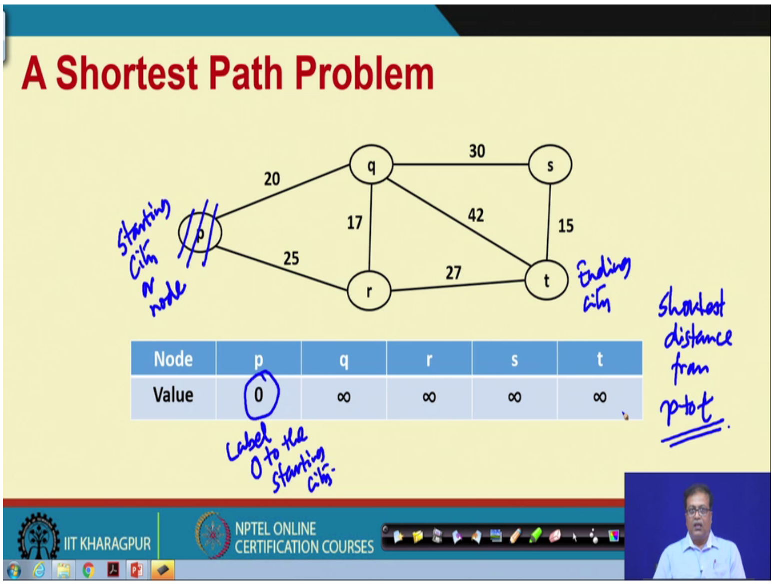
(Refer Slide Time: 01:00)



The reason being that those algorithms are good, but Dijkstra's algorithm is elegant and simple, and it has got one additional advantage right. So, what is that advantage I am going to discuss very soon?

So, as you can see in this slide the Dijkstra's algorithm is applicable for single source shortest path problem right. Single source here basically mean that, if we have only one source where you are starting. So, this algorithm is based on the concept of dynamic programming; that means, it breaks the problem into a set up of sub problems and then combine those sub problems solutions to a larger problems solution, but it is not really a divide. And conquer kind of thing, it is basically finding out the relationship between sub problems. And you know through one sub problem solutions; find out the next sub problem solution stage by stage.

(Refer Slide Time: 02:08)



So, you know the way we would like to solve that we take a simple problem first and solve that problem, without really introducing the algorithm. And from the knowledge that we gain from solving a simple problem, use it you know then show the algorithm and then follow the algorithm ditto you know and show another example right. So, that way understanding the algorithm will be easier. So, this is and basically problem a shortest path problem.

So, if you look at the kind of shortest path problem that we have taken earlier, these cluster of cities in those cases you know earlier when we taken. So, supposing I think of it in the conventional manner, then if you look at then this is one cluster of cities, this is another cluster of cities and this is another cluster of cities.

What we did actually? Moving from one cluster of cities to the next cluster of cities to the next cluster of cities, we use to divide the problem into stages. That is what we did in all the earlier problems that we have solved. But here there is a difference; if you really you know remember then in the kind of problem that we had solved, there was no path between a cluster of cities.

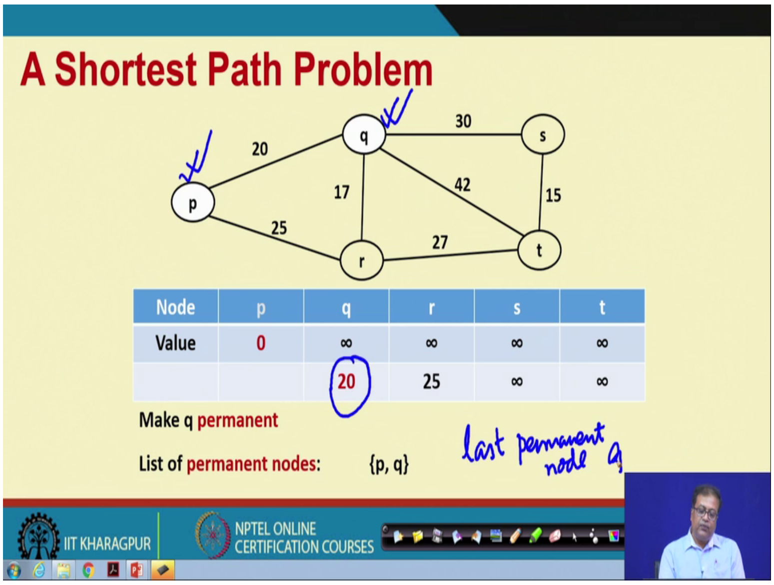
So, in this case q and r suppose they are would not have been any path, then those methods will be very much applicable. But supposing we also have a path between q and r or in this case there is a path between s and t is it alright. So, we have a path from s and t as well. So, if you have that and also the path between q and r, you cannot really say that these are one set of cities, these are another set of cities. So, go from these set of cities to be set of cities to the set of cities and things like that is not possible. In fact, you know in such situations, you have to deal probably every city at a stage is it not going from a given city to another city as a stage is it alright.

So, now question comes that, you know which one is the least path to a city from another city is it alright. So, because this is network, you know it is a set of connections that there between the cities. So, that cluster approach that we have followed earlier, may be little difficult here is alright. But then this simple algorithm that is Dijkstra's algorithm really helps us in solving this problem very simply. So, the method let us see what is that methods? So, what we do in the very beginning with every node, we assign a level right. These levels could be like this, that the starting city. So, look here this is our p is our starting city or node.

So, p is our starting city or node. So, we assign a level 0 with that right a level 0 to the starting city right. So, what we really want to find out is? The shortest distance from p to t right, so this is the starting city and this is the ending city. So, we want the shortest distance from the starting city p to the ending city t, and we want the shortest distance. So, what we begin the Dijkstra's algorithm in this manner, we put a level 0 to the starting city and you know level infinity to all the other cities that is our stage 1 is it alright is it clear to you.

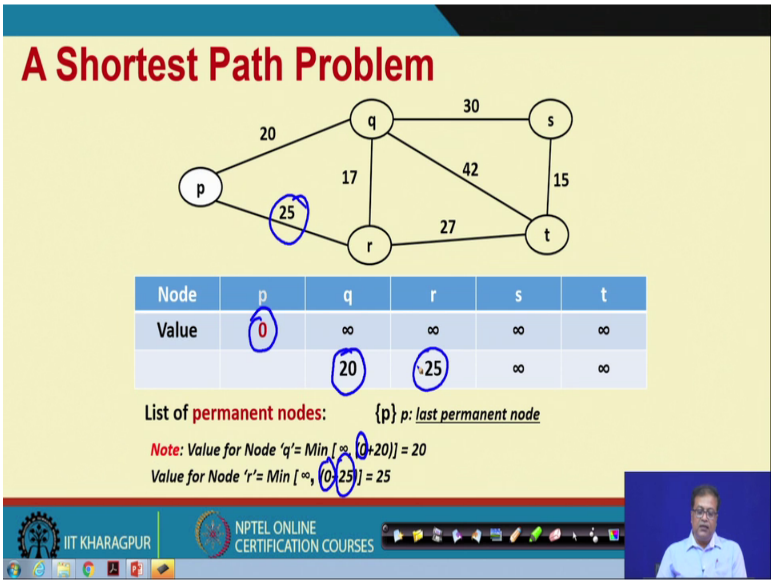
So, if it is clear, then next let us see what happens to our next situation.

(Refer Slide Time: 06:51)



So, now, p our starting city we make it permanent right. So, we have made p permanent. So, if you see we have just coloured it is you know white. So, p is permanent and put red colour here; that means p is permanent. So, at this point what are our permanent nodes? Only one; that is p right. So, that is what we have done that, we have started from p put a level 0 to that p, and made it permanent. So, p is our permanent node.

(Refer Slide Time: 07:33)



Having done that let us go to the next slide, now at this point the our list of permanent nodes includes only one city that is p. So, we let us call p is our last permanent node.

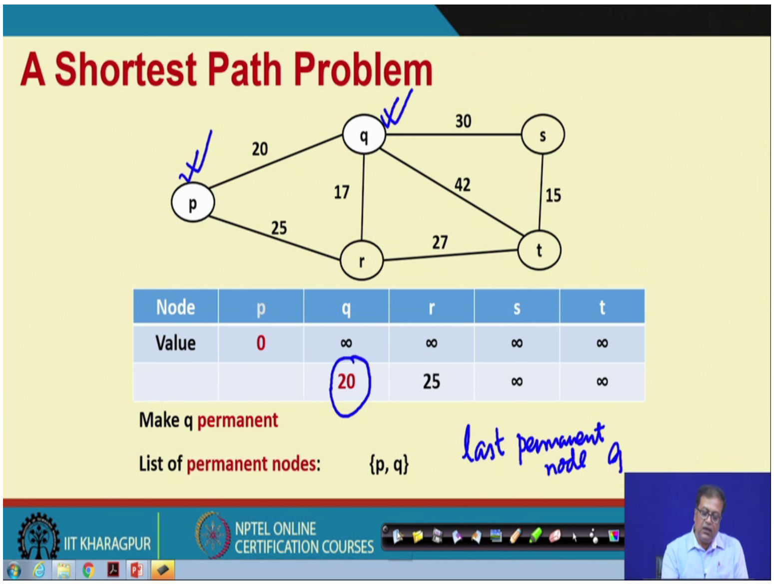
Now you aprobably understanding the stage concept here; so, the stage that we have already gone through is the p. That means stage 1 is over; now we are in the second stage. But we really do not know which city we are going to move now, is it alright because the cities are not in clusters right we can go from any city to another city because, this is kind a way network situation. So, because it is a kind a way network situation so, you know we can go from any city to any other city.

So, what we do now, we see from the city p, what are the city to which we can go to directly. So, can you find which city we can go directly from p, you are sure that you know we can go either to q or to r. So, if we go to q, the q we know the value for node q will be the minimum of its current level that is infinity and the last permanent node p to q whatever distance and that distances is 20 is it all right. So, therefore, the value for node q will be minimum of the current level value plus the last permanent node value which is 0, this 0 is it not.

So, this 0 last permanent node value and this 0 the same 0 is here, plus the distance from p to q which is 20. So, the minimum of infinity and 20, it becomes 20 and that 20 is written here all right. So, you can easily know that what will be the value for r it will be minimum infinity 0 that is p 0 plus 25, this 25 is here. So, this should be 25. So, this 25 is obtained is it alright. So, this is very simple calculation really.

So, like this we compute from the last permanent node whatever nodes we can reach, we actually highlight them. So, when we have highlighted these nodes, now we move over to our next slide and then we find out that, out of the values which are remaining which one is the minimum. As you can very clearly see q is the minimum value. So, q we highlight and make q permanent.

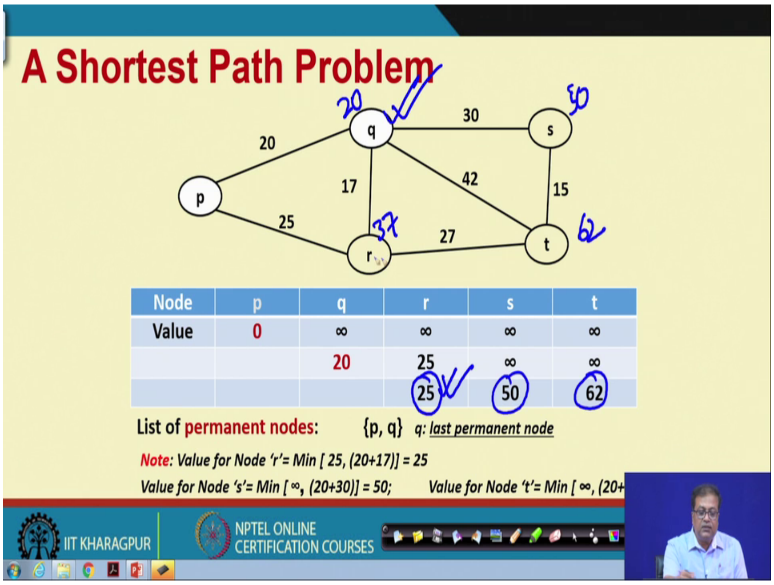
(Refer Slide Time: 10:48)



So, you can see here that q is now highlighted. So, q is permanent and what are some permanent nodes now? The permanent nodes are p and q. Which one is our last permanent node? Out of p and q because these two are permanent nodes, which one is our last permanent node? Last permanent node is nothing, but q is it ok. So, last permanent node is q this fact we shall make use and we shall see that, go to our next calculation. So, q is our last permanent node now from q again we can compute r. Because q is path with r with s and t. so; that means our second stage is over now we have come to the third stage.

So, if you now have in the third stage the options before us really are from q, there is a direct path to r, there is a direct path to t, there is a direct path to s. So, if you really compute those values, then r which becoming the minimum of 25 and 20 plus 17 that is 37, the value is nothing but 25 itself, so, there is no change it remains 25, is it alright. When it comes to s the value is infinity so minimum of infinity and last permanent node is q so last permanent node was q. So, this was last permanent node.

(Refer Slide Time: 12:45)

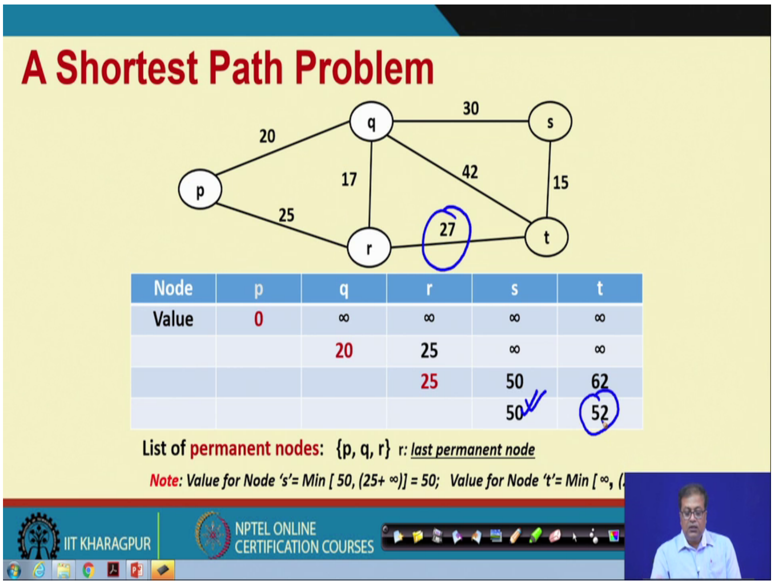


So, using that we may calculate this value was 20. So, these value will be 37, these value is 62 and this value is 50. But then out of that current value is 25, and 25 and 37 if compare, then 25 is lower the minimum. So, I leave at 25, but out of infinity a 50 is lower. So, I put 50 and out of infinity and 62, 62 is lower. So, I put 62 is it alright? So, these values we can compute. Now you will tell me which node we shall make permanent now? You know the remaining values are 25, 50 and 62 right.

So, 25 is minimum, so r will be minimum very clear then so, r will be made permanent, but r is made permanent from where? So, this is another very interesting point, r is the new minimum new node, which is made permanent, but r is not made permanent from q, r was made permanent from p itself that we should remember, is it not. So, r was made permanent from p, when we saw the q distances nothing much changed. So, q is in discarded as for as r is concerning that sense. So, r was made permanent from p; when r is made permanent it was made permanent, because r was made the value was calculated from the value of p, that we should remember right.

So, now come to the next slide, now we have made r as permanent. And if we make r as permanent the last permanent node also will be r. So, at this stage these are the values and p q r are made permanent.

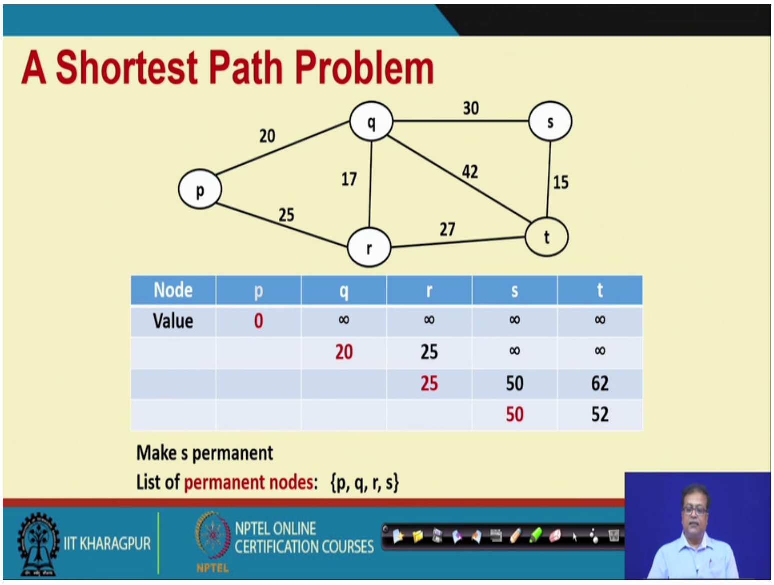
(Refer Slide Time: 14:34)



So, that means, now we move over to our next slide and in our next slide the last permanent node is r. So, we see the distance from r what is the distance? Only one distance is available it is not. So, from r only one distance that is this one. So, that was 27. So, if you add that 27 to the value of r 25, how much you get? 52 is it not? You can a also compute s, but then there is no distance from s there is no distance so; obviously, nothing will be computed that we remain 50 itself right.

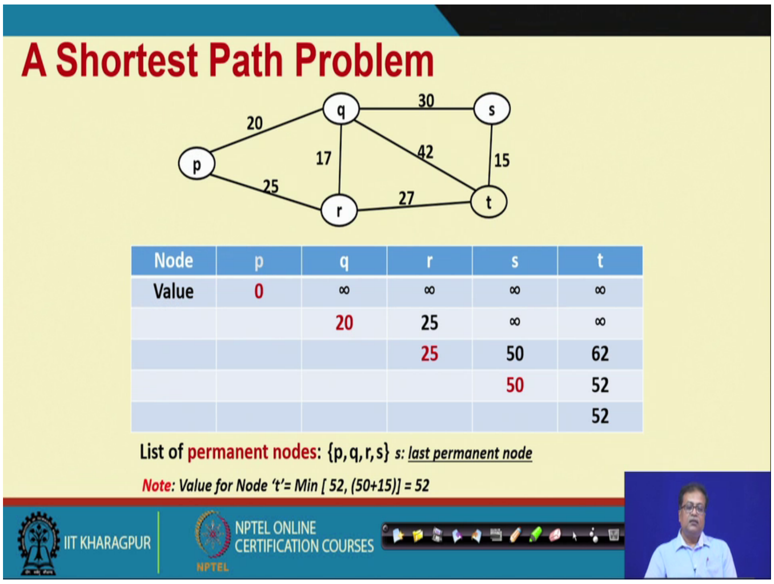
So, this 52 is computed now and this 50 was whatever the old value because r 2 has no path. So, nothing could be computed. So, 52 is you know we already had a value of 62 to t, but this 52 is lower. So, if you remember that t is made permanent from t is computed from r right. So, if you remember that. So, this value is then 52. So, once you have these values, now again let us move to the next slide.

(Refer Slide Time: 16:05)



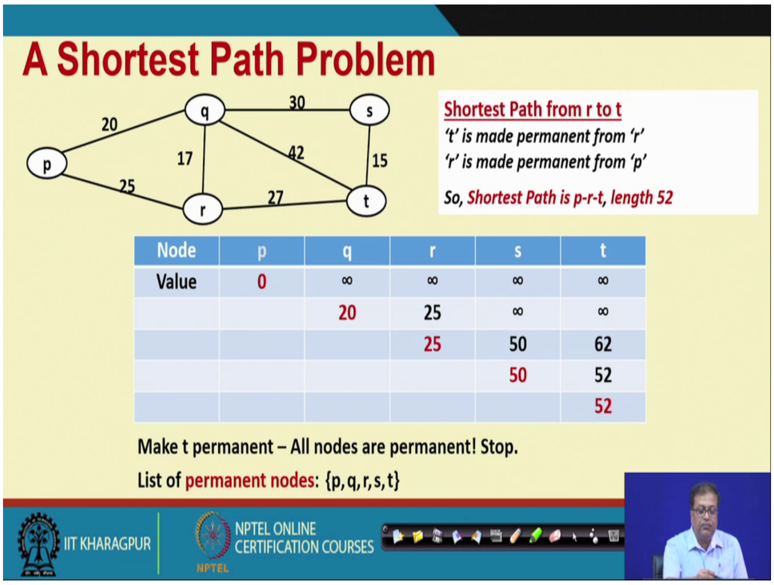
Now next slide out of this 50 and 52, 50 is lower. So, we make s permanent. So, now, we have p q r s all are permanent. So, come to the next slide.

(Refer Slide Time: 16:17)



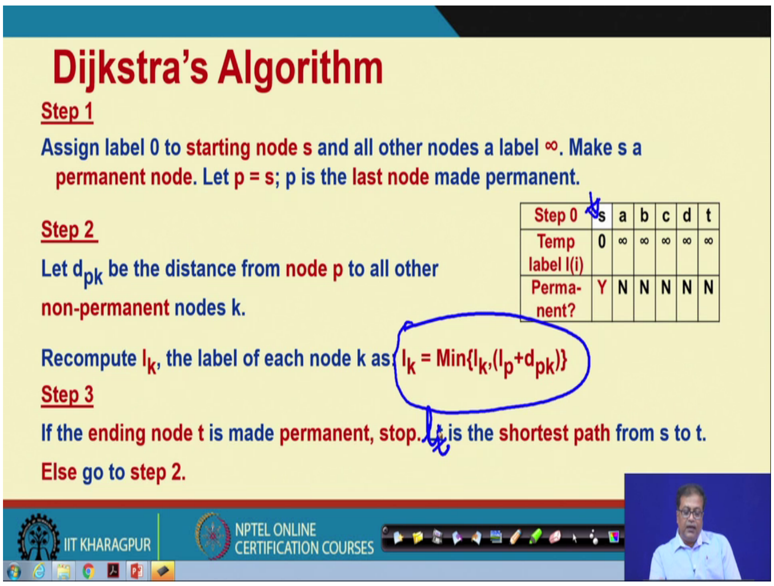
Now last node to be made permanent is s. Now s to t distance is 15. Now value for s was 50. So, it is the minimum of 52, and 50 plus 15 there is 65. So, 52 and 52 is lower so; that means, t value will become 52 only it does not change is it alright.

(Refer Slide Time: 16:46)



So, now we have got all the values computed and all nodes are permanent p, q, r, s, t, to be specific t is permanent. Now if you look at t was made permanent from r if you recall, and r was made permanent from p right. So, therefore, the shortest path from p to t was p r t and that value is nothing, but the 52 alright. So, shortest path is p r t and the length is 52 and how it was found? Because if you re call that that value 52 was obtained from r and the for r s value was found from p therefore, p r t would have been our new shortest path. So, this is how the Dijkstra's algorithm really works. So, having understood that, let us now see the Dijkstra's algorithm.

(Refer Slide Time: 17:47)



The distress algorithm has got basically three broad steps. The step one: Assign level 0 to starting nodes s and a level of infinity to all other nodes. And then make s the starting node a permanent one. And let p equal to s what is p? P is the last node that is made permanent. So, then we make a small chart like this that step 0; s, a, b, c, d, t these are all the nodes the temporary level 0 infinity all those are infinity permanent; yes Y.

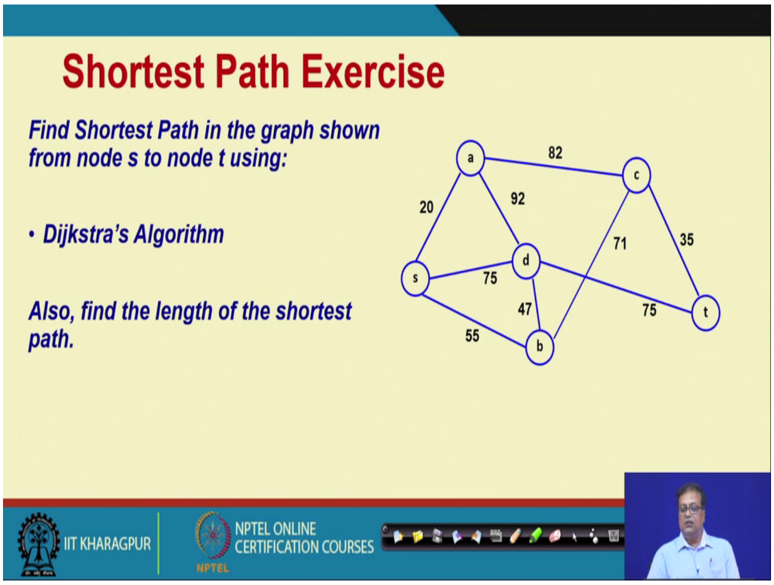
So, this is permanent which one is the last node to have made permanent? That is s right. That is what is the first set of calculation; then what we did the step 2? Let d p k be the distance from node p; what is node p? That is a last node that has been made permanent and what is k? K are all the non permanent nodes is it alright. So, recompute l k the level of each node k as l k equal to minimum of l k and l p plus d p k is it all right. So, l k equal to minimum of the currently available value that is infinity or whatever, and l p the value of the last permanent nodes say 0 here plus the distance from s 2 that value.

So, if it is a, then the l p is 0 because l p is l s which is 0 and d pk is d sa whatever the distance is a s 2 a that you have to add. So, like this you proceed and finally, in the ending node t is made permanent a stop right. So, the shortest path is you know a will be the value that you have got the; that means, the l t which is the shortest path is l will be not capital this will be small. So, l t will be the shortest path from s to t, else go to step 2 all right. So, I will skip on computing

So, I think a Dijkstra's algorithm is clear to you now that start with a temporary levels made one of them the lowest one as permanent. In the first case because they are all infinity and only the starting node is 0; that means, make the starting node permanent and from the last permanent node find out the distances and find out the minimum the which one is the minimum you make it permanent is it ok. So that means, you know a series of you know as you are re computing l k's the different levels you keep on making further nodes permanent based on whichever is minimum at a given point of time. Is it alright? That is not written, but that is what is to be done.

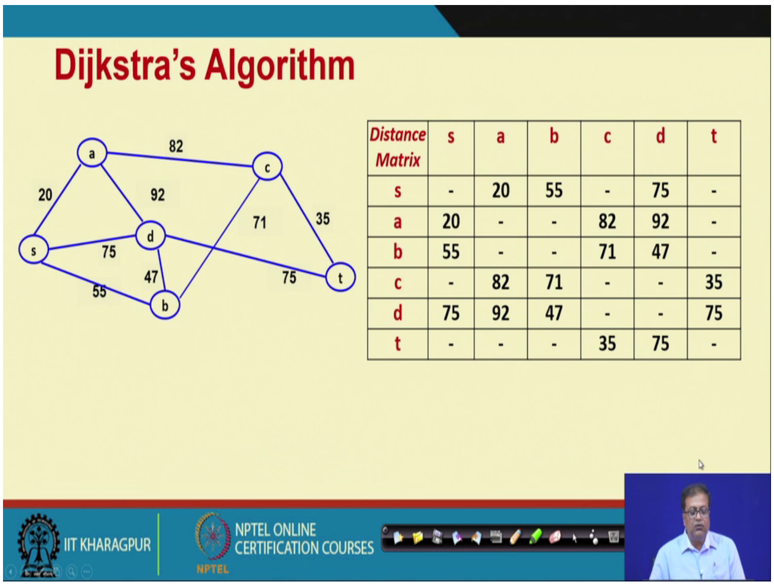
So, once that is done then let us see one more example of how all these things happen.

(Refer Slide Time: 21:15)



So, let us take this shortest path example, so find shortest path in this graph shown from node s to node t using Dijkstra's algorithm, and also find the length of the shortest path. So, this is the graph that we have from s to a to b to c d and t. So, used Dijkstra's algorithm to find the shortest path and also find the length of the shortest path.

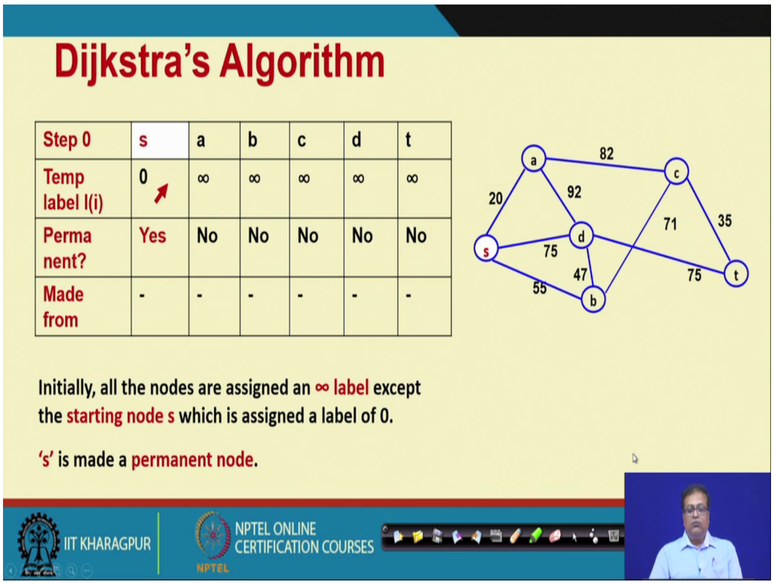
(Refer Slide Time: 21:48)



So, let us see how it is computed. So, initially usually sometimes you know that you may not be given the graph; you may actually be given this distance matrix. So, you know actually this is one of the way by which this graph can be represented. So, you write s a b c d t both sides and use put their distances.

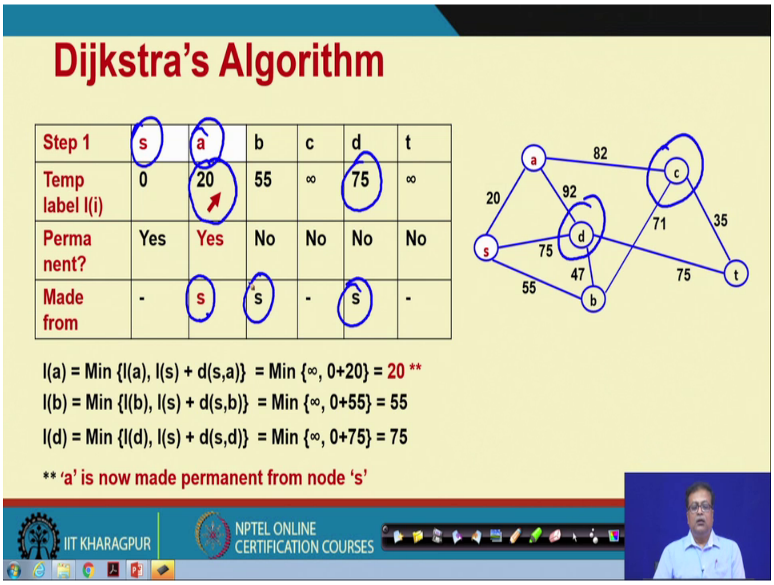
So, you see in this case the distance from s to a; and a to s is same, but sometimes they may be different also right. That going from a city to another city and coming back from that city to those others first city may not be always same, you know if that could be a other situations as well, but we are not going to discuss them here.

(Refer Slide Time: 22:37)



So, how do we begin? How do we begin is that, first you put some temporary levels that is 0 infinity, infinity all those and 0 here and make s permanent and all the other are not permanent, we have one more thing call made from right. So, if we also keep that made from, then the path computation will also be very easy. So, initially all the nodes are assigned an infinity level except the starting nodes s which is assigned a level of 0. So, s is the made the permanent node. So, that is where we begin.

(Refer Slide Time: 23:36)



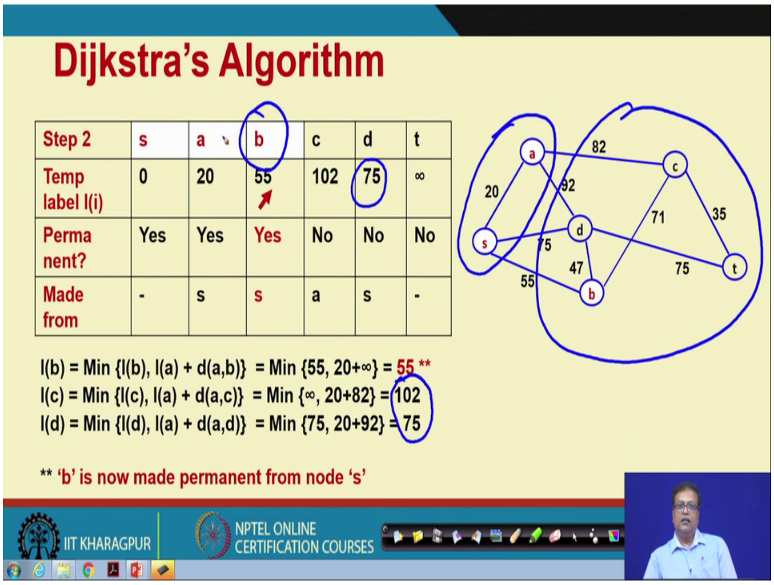
So, s is made permanent, so next see the next step what happens. Now from s we have path to a to b and to d. So, we can compute a b and d. So, as you can see the distances so, either through the graph or from the matrix, we can find out s to a, s to b or s to d is it all right. So, we can actually find out s to a, which is you know 0 is the last node. So, 0 plus 20, 0 plus 55 and 0 plus 75, their current level was infinity. So, out of that obviously, this 20 55 and 75 they are minimum, so we put them. So, 20 is the minimum.

So, we highlight 20 and you make that 20 as permanent right. So, we make a also permanent, s was already permanent, now a is the last node to have been made permanent so, make a permanent. So, you should tell me that once I have made a permanent which node should be you know recomputed now? What are those nodes which will be re computed now, which one is the last node to have been made permanent? Obviously a because a is the minimum value that we have got from s and we made it permanent.

And then from a, we find distances from to which nodes? Look at this graph there are two nodes one is your c another is your d where you have some path. D value is already there that is 75, the value of a is 20. So, if you add the distance 92 with it so, it this new value will be 112. So, 112 and 75 which is already there. So, do you think the value will be changed? The value will not change the value will remain at 75 is it alright? But 20 plus 82 the value of c which was infinity; obviously, it will change and it will then become 102, I hope you understand.

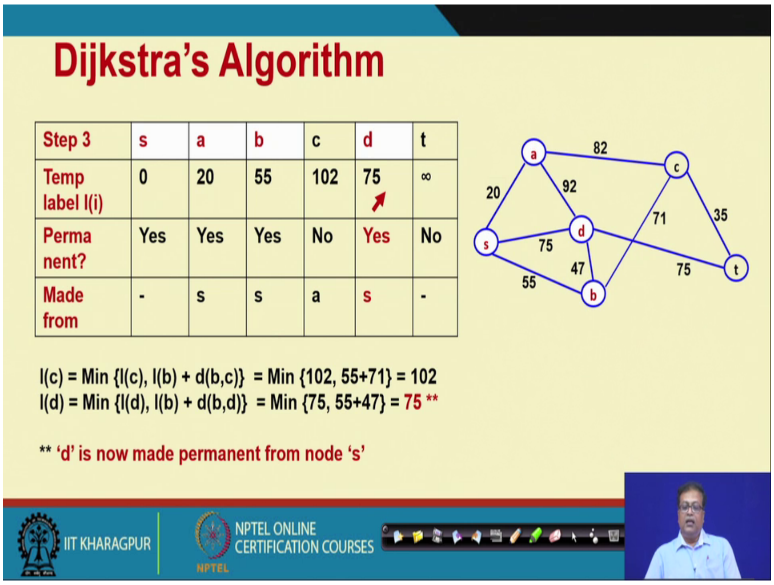
So, one more thing to note that, you know when we computed those a and d you know you have found out that a has been made from s and d has also be made from s right. So, we noted that and b was made from s as well. So, all of this has been noted. So, after having noted this let us move to the next slide and here we find that, we have you know made b permanent now, and you know after making sorry after computing a that is 20, then we have found out that the 3 other values that is in you know the lb was already there because there was no path to b. So, it remains 55.

(Refer Slide Time: 26:23)



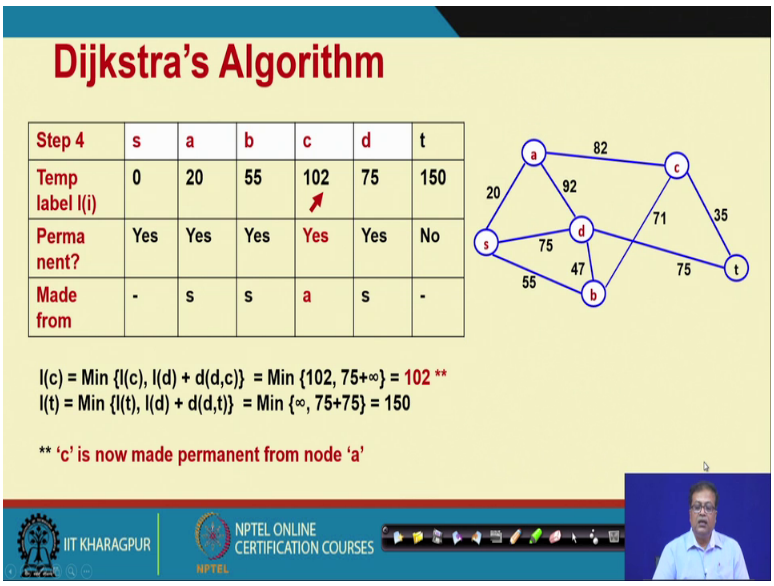
But then we will compute c which was 102, and when we computed d we says that you do not change it because out of 75 and 112, 75 remains, so that 75 remains so at this stage. So, we have calculated really 102 and 75, we have not calculated this 55 and a to b there is no path also. But then after having computed these values we find that out of the remaining because earlier we had s and a permanent, this two are permanent. So, now, we find that out of the remaining 4 values. If we see the two different classes, which one is the minimum? The minimum is 55. So, since 55 is minimum that is how we make b permanent is it ok. So, now, b permanent and then we see further distances from b. So, how do we do that let us see.

(Refer Slide Time: 27:55)



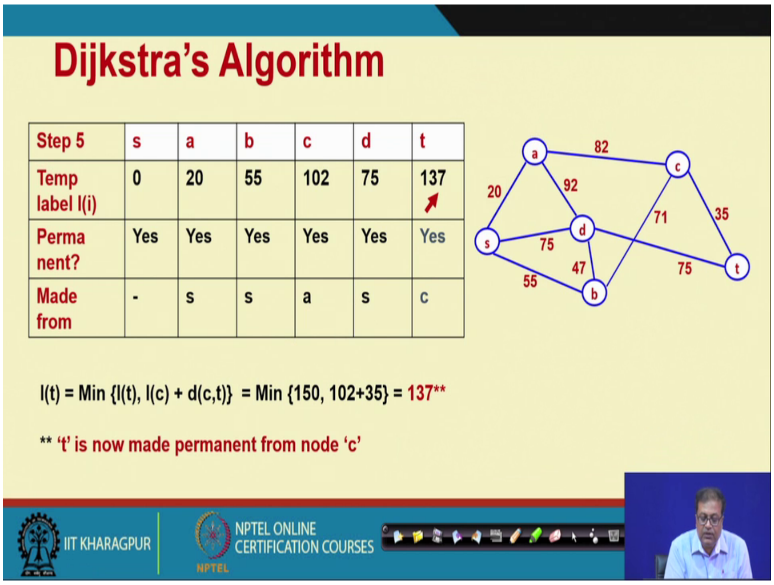
So, again we will come to the next slide, and from b we find some distances from b we find distance up to c and distance up to d. So, you see the distance of up to c was 55 and 71, but 102 was already there, 102 was lower. And the other value again 75 is lower than this value. So, we find that we find the 102 and 75 so that will be the new computation and 75 is lower. So, I put d and d is the permanent node now. So, we have now s a b d these four are permanent. So, again from d we see distances, only one distance is there that is t.

(Refer Slide Time: 28:46)



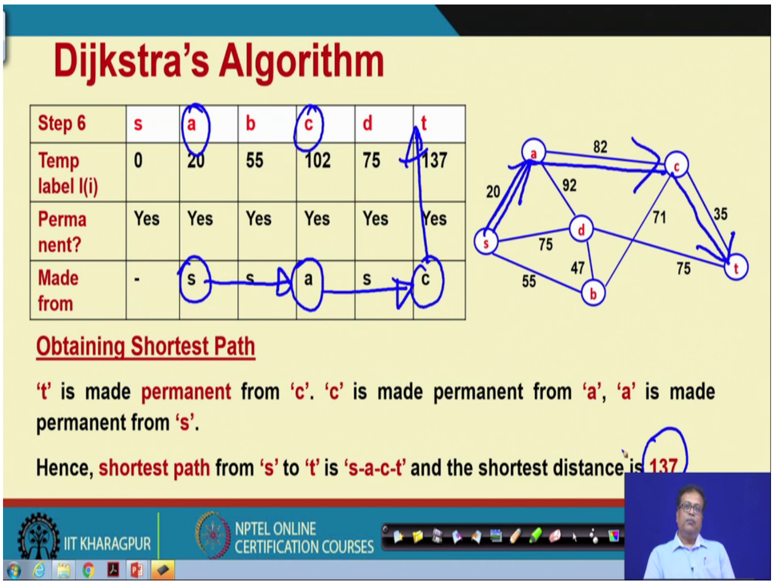
So, in our next a value we again computed and this time I find the d to t, a value of 150. So, we have put a value of 150 to t and for c, we have a value of 102 because there is no path from the last permanent node one that was the d. So, if we have done that then this time we make c as permanent. So, c is the last node to be permanent. Now again from c, which is 102 we have a path to t.

(Refer Slide Time: 29:24)



So, this time when you compute then we find that between 102 and 35, 137 and 150 which is already there, 137 comes out to be lower. Since this is lower this value is put and then t has been made permanent from c is all right. So, now, our entire thing is done all the nodes are made permanent, at least t is made permanent. So, when all of these are made permanent. So, we can find our final table.

(Refer Slide Time: 29:53)



In our final table what we find is that these are the all the temporary levels are now permanent and this is how they made from. So, t is made. So, now look interestingly t is made permanent from c, c is made permanent. So, c is made permanent from a; a is made permanent from s right.

So, what is our path? Path is s to a and a to c and c to t right. So, s a c t is our shortest distance. So, this one s to a; a to c and c to t right that will be our shortest path and that shortest path of distance in this case will be one 137 is it alright. So, this is how the Dijkstra's algorithm is used and that is how we shall compute these in such kind of problem. So, this is a very simple algorithm and this algorithm is based on dynamic programming again right. So, we have seen different types of dynamic programming and how do you solve them. We conclude our dynamic programming here. Next week we shall begin integer programming right.

Thank you very much.