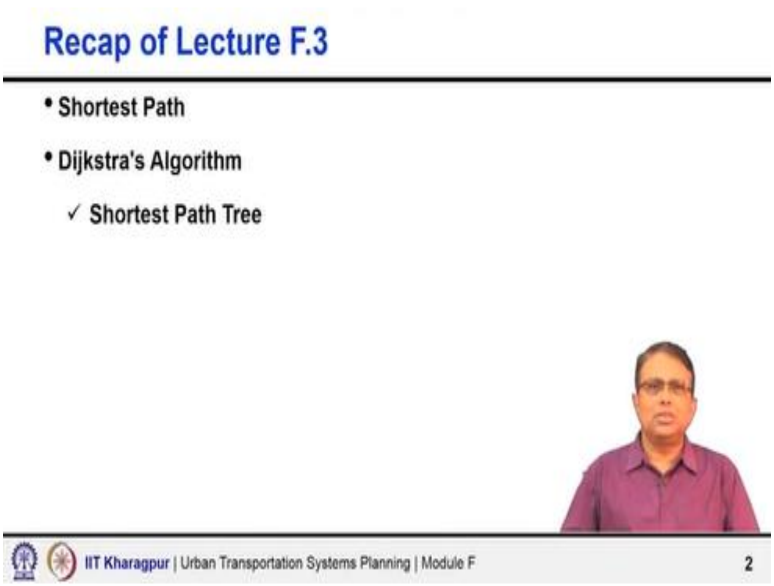
Urban Transportation Systems Planning
Prof. Bhargab Maitra
Department of Civil Engineering
Indian Institute of Technology, Kharagpur

Lecture - 44
Network Algorithms-III

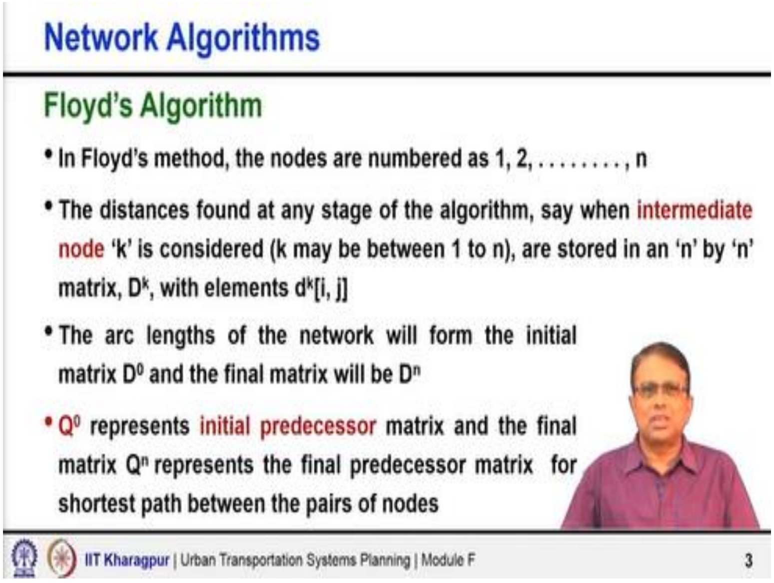
Welcome to module F lecture 4. In this lecture we shall continue our discussion about network algorithms which are related to traffic assignment.

(Refer Slide Time: 00:26)



In lecture 3, we said why we need to identify shortest path in the context of traffic assignment and then one very popular algorithm which is known as Dijkstra's Algorithm that we discussed. And also, I took an example of a network and then explained you step by step how you can apply Dijkstra's algorithm to find out the shortest path from one home node to all other home node. Now in continuation to our same discussion regarding the; shortest path how to find out the shortest path.

(Refer Slide Time: 01:14)



Today we shall discuss about another algorithm, which is also very popular known as Floyd's Algorithm. This algorithm is generally I can say more efficient because what we wanted or what we were required to do say, if there are n number of nodes in a network to get shortest path from one home node to all n home nodes, we wanted n iterations in the Dijkstra's algorithm.

Here also it is iterative and we have to follow the iterative approach to get it. But in one round that means in n iteration we can get shortest path from all home node to all home node. That is the beauty. So, in Floyd's Algorithm the nodes are as usual numbered as 1, 2, 3, 4. Let us say there are, n number of nodes and the distance found that any stage of the algorithm say, when we are considering every state, we consider an intermediate node.

I shall explain you off course in much more detail manner with also an example. But every stage we shall consider an intermediate node and that intermediate node we shall start with considering number 1 and then every stage will increase it by 1, so then number 2 number 3. So, if I say generally, we are considering a node k where k may be any number between 1 to n depending on what is the iteration number.

This distance are stored in an n by n matrix because as I said we are dealing with shortest path from all home node to all home node. That means 1, 2, 3, 4 like that if there are n node from each of this home node to all other home nodes, we are trying to find out a shortest path. So, we are actually storing that distance data in a matrix called D k indicates that it is the matrix at the kth iteration.

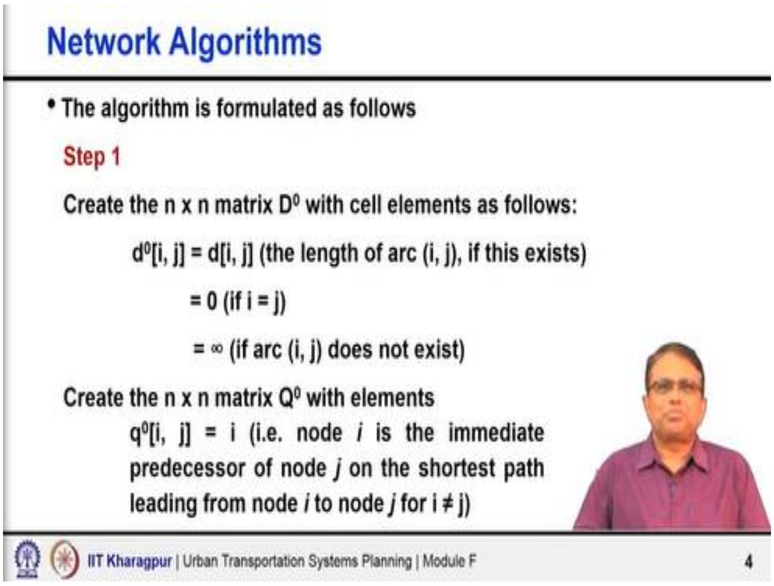
With elements D k i, j it is a 2 dimensional matrix from node i to node j we are finding out the shortest path so these values are stored. And the arc length of the network will deform the final matrix the final matrix will of course be coming out at the end of nth iteration. So, that will be D n and the initial matrix whatever we are going to assign in the beginning that we call it as D 0 that is the initialization.

And then we keep updating these matrices and when they are required or contextual. And finally, by the time we go to nth iteration, whatever will be the D n matrix that is the final shortest part. And remember that I said that in every, iterations we are considering an intermediate node. So, every node is actually considered as an intermediate node. Similarly, the initial predecessor also we shall store in again n by n matrix.

But that cell will denote what is the predecessor? For example, if I consider a cell 1 to 5 so let us say we are considering about from 1 to 5 then this cell value will indicate what is the immediate predecessor node of 4 that node is stored. Suppose predecessor node of immediately 4, 5 is say let us 3 then we shall take 1, 2, 3 that cell will take what is the predecessor node for 3.

Like that we can back trace and finally we can come back to the home node, whatever it may be in this case 1 or maybe from 2 to another node we are talking accordingly would be. So, there also every iteration as we are updating the distance matrix wherever any cell we are updating in the distance matrix, then we are also updating the corresponding cell particularly the predecessor in the Q matrix. So, Q also will finally have Q n which is the final predecessor matrix.

(Refer Slide Time: 06:08)



Now how to go about it? How to apply this algorithm? We shall apply this algorithm step by step or again, as I said it is an iterative approach. First, step one is kind of initialization. How we are doing the initialization? We have n by n matrix, that matrix we are calling as D 0 matrix and there will be elements d 0 i, j from node i to j the shortest path we are talking about. That will be equal to d i, j if arc i, j exists.

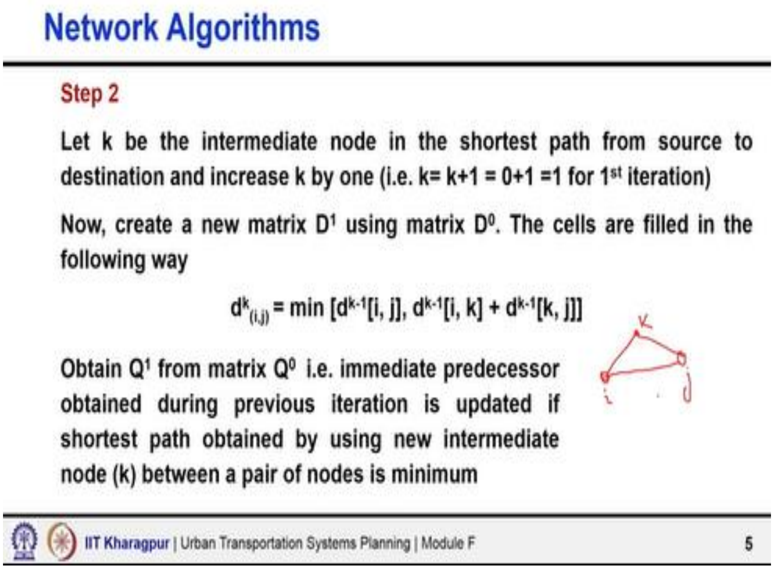
Suppose 1 to 3 we are talking if 1 to 3, there is a direct connection then whatever is the value we know that value we shall put here. It will be 0 if i equal to j. So, all the diagonal elements will be 0 1 to 1, 2 to 2, 3 to 3 all will be 0. And if there is no direct connection suppose 1 is not connected directly to 3 in my network, then I do know what is the shortest path. So, I will assume that this is infinity, the length is infinity.

Infinity means it is I am actually assuming a very high value. You can as well assume a very high value, which is higher than any of the, you know length of any of the link in the network. So, with that that way we create that D 0 matrix and I explain how we consider the element. Now in the similar manner we also create a n by n matrix which is predecessor matrix and since this is the initialization that will be Q 0.

Now how this element or how we consider this element we will say q 0 i, j = i. Because what we explored in the distance matrix suppose I considered home node 1 so I said 1 to 2, 1 to 3, 1 to 4, 1 to 5, 1 to 6 like that up to 1 to n and in each case, I took the length of the link which if there is a direct connection if there is no direct connection then we do not know so I assume the cost has infinity.

That means all these one to all other node the for all the nodes target nodes or destination node the predecessor is i so that is what it is. So, first what will happen? One to all other node all predecessor is 1, 2 to all other node all the predecessor value for each destination cell is 2, 3 to all cell 3 like that. So, that is what is my initial distance matrix and initial predecessor matrix. Now with this we go to the next step, step 2.

(Refer Slide Time: 09:08)



Now let k be the intermediate node in the shortest path from source to destination and now we shall increase k by 1. So, k was initially assumed Q 0 D 0. So, k was also assigned initially as 0 value. I could have mentioned in step 1 also but I mentioned in the beginning of step 2 and then now we are going for iteration. So, I will increase k by 1 value. So, k will be taken as k + 1.

So, when I am going for the first iteration what will be my value of k? Value of k will be 0 + 1 k. When I go to the second iteration, then what will be my value of k? k was already 1 so now it will be taken as 1 + 1 2 like that. So, whatever will be the k value k is since it is 1. Now, we are definitely trying to get a distance matrix which will be the D 1 matrix and also, the Q matrix which will be the Q 1 matrix.

So, how to get the D 1 matrix? Generally, I am saying if I am trying to get the d k i j k in the first iteration is obviously 1. But generally, d k i j is minimum of carefully observed you can understand all the things very easily, minimum of 2 values, what are the two values? So, far what is my value? My value is d k - 1 i j in the previous iteration whatever was my best value.

In this case, if I am the first it if I am going for the first iteration then what was the d 0 i j that will be the fourth iteration, it will be third iteration d 3 i j fifth iteration, it will be fourth iteration value. So, whatever till now whatever is the value best value, I have taken I am comparing that with another value. What is that another value? I am saying, d k – 1 i, j i to k and d k - 1 k j.

That means earlier if I try to explain you further say, I have i here, I have j here. So, I said what is my best i j? Till now we got in the last iteration and done then I am bringing one iteration k and I am saying whether d i k + d k j. This k is the basically intermediate node that I am currently considering this is i and this is j. So, if I find still my d i, j whatever was there in the previous iteration, that is the shortest path.

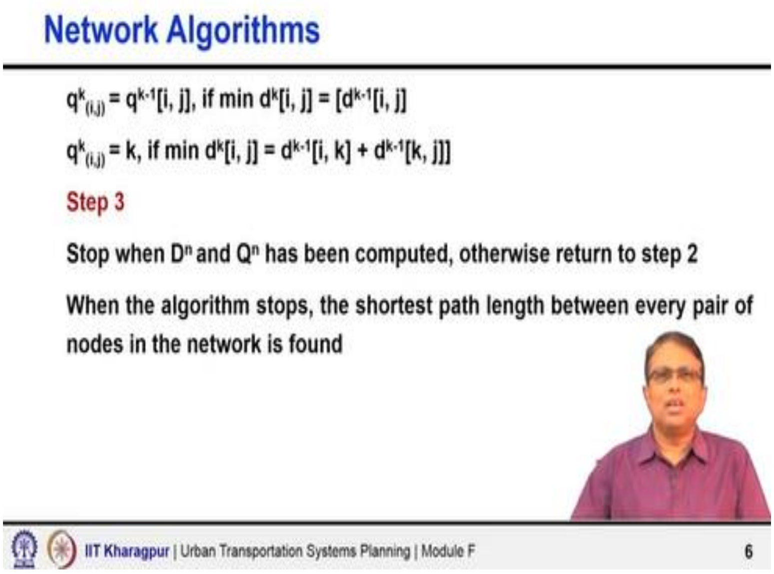
Then by traveling from i to k and then coming from k to j, I am not able to reduce my distance, so I do not update the value whatever was my d k-1 i j that remains as a value in my d k i, j but if I find that d k - 1 i k + d k - 1 k j. That means traveling through k instead of traveling from i to j whatever may be the present update we got there may be direct may not be direct also but what is the best value so far, we got.

I am only exploring that value is better or I can travel from i to k and then come back from k to j that is better. Now if the second one is cheaper then what I will do? I then need to update my matrix, so now then my d k i j will be d k - 1 i to k + d k - 1 k to j. Addition or the total distance, that will be my new shortest distance from i to j as far as we have explored and till k the direction that is the best value.

Now again if the first one is taken that means d k - 1 i j, I find that my earlier solution is better than the predecessor matrix also I do not update because my predecessor matrix also does not change. But if I change or accept the letter value that means d k - 1 i k + d k - 1 k j. That means I am now not travelling directly but I am now coming via k of through k. So, my predecessor node for j becomes what k instant of the present predecessor.

That presents are also may not be i because that may be in later iteration first iteration, it will be i but in the later iteration if I have explored another intermediate node better. Then my predecessor value is stored there, that node but now whatever is my k value I will update it.

(Refer Slide Time: 15:15)

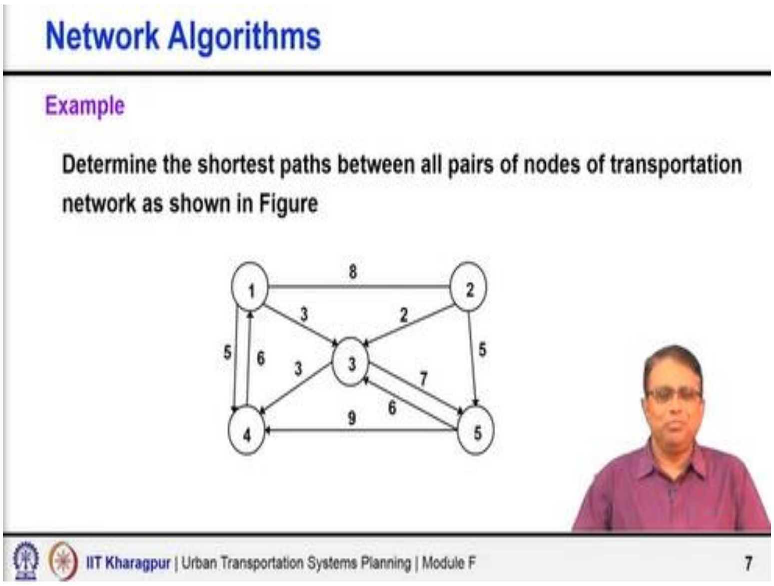


So, with this that is what I say my q i, j will be q k - 1 i j if minimum d i k equal to d k - 1 i j. And if I take the latter one that second part becomes the cheaper then I change the predecessor value and I make the predecessor of j as k. See how intelligently it is done or how efficiently it is done, now k value is changing from 1 to n. Can you just imagine what is happening actually?

So, iteration to iteration first suppose say 1 to 7 or 1 to 5, so what I will do? First time checking if there is a direct connection 1 to 5, I am saying, that is my best. Then I am checking 1 to 2, 2 to 5. Then in the next iteration I am checking 1 to 3, 3 to 5 then in the fourth iteration I am checking 1 to 4, 4 to 5. Like each intermediate node I am checking whether travel via that or through that intermediate node whether I am able to reduce my distance.

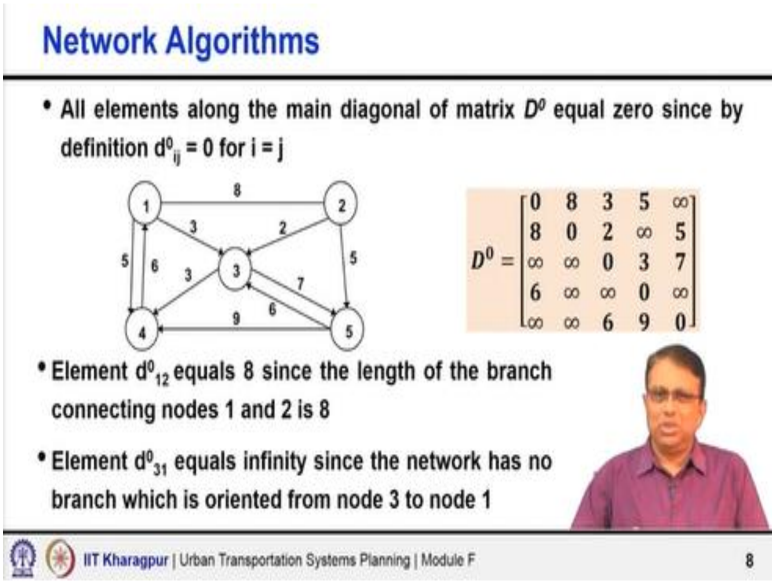
If I am able to reduce my distance then I need to update my matrix if not then the same solution I shall keep. And if I am updating if I find that travelling through this new node, what is the immediate node k is actually I am able to reduce my distance. Then I also need to update the predecessor matrix that is all very simple.

(Refer Slide Time: 17:03)



Now let us take an example. You can see this network. And I shall show you how you can apply Floyd's Algorithm to get your shortest Matrix and final predecessor matrix.

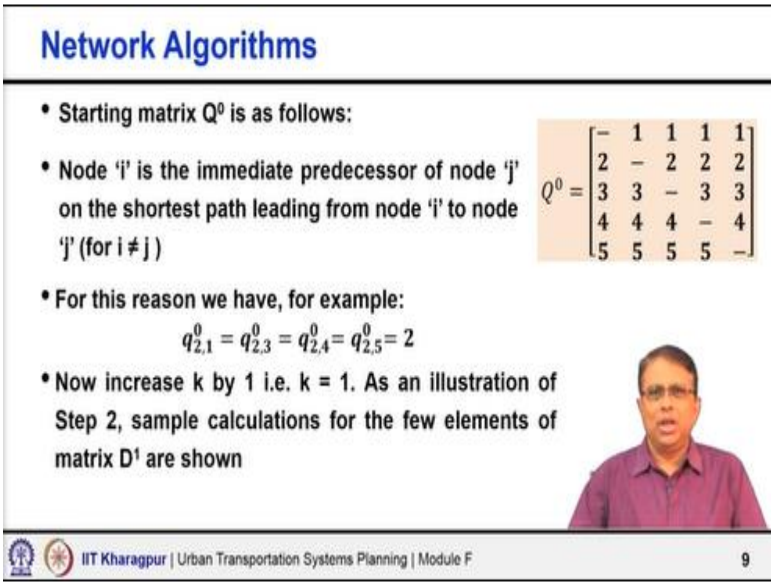
(Refer Slide Time: 17:22)



Let us first go with d 0. So, let us say 1 to 2 you know, there is a direct connection so 1 to 2 value is 8, so I have put it 1 to 2 is 8, 1 to 3 also there is a direct connection 3. So, I have put that value 3. 1 to 4 also there is a direct connection the value is 5, so I have put the value of 5. But, 1 to 5 there is no direct connection you can see so that cost I do not know. So, let us I want to assume a very high cost infinity.

Similarly, all other cells you can fill up say 2 to 1 8 2 to 2 obviously diagonal elements all will be 0, 2 to 3 2, but 2 to 4 again, there is no direct connection so it is infinity. And 2 to 5 there is a direct connection so value is 5 like that I follow for my D 0 matrix initial matrix.

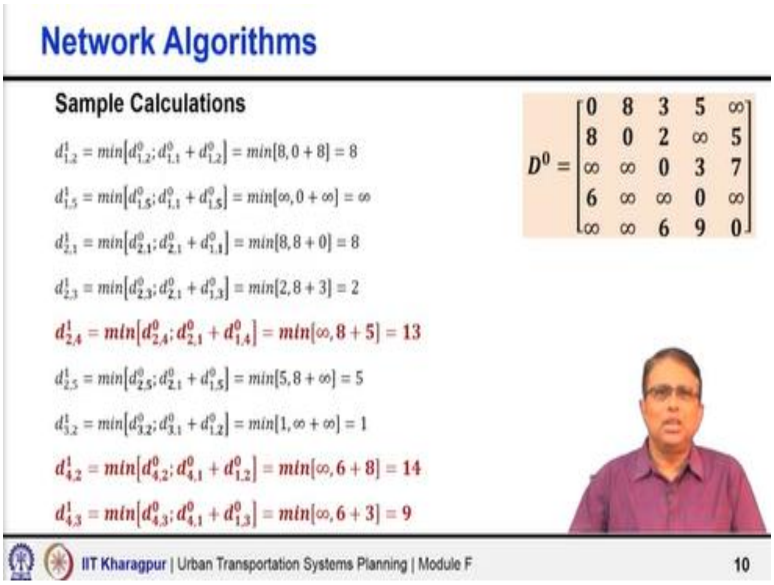
(Refer Slide Time: 18:25)



Then what I do. I have to also make my q 0 matrix predecessor matrix you see from home node 1 to all other node 1 all sense 1 from 2 to all other destination nodes 2, 3 to all other destination node 3, 4 to all other destination node 4, 5 to all distinction node 5. Because my distance matrix is based on that, you know, every time every row we take with that origin point of that home node.

So, that is what is my predecessor, so, I am ready with my D 0 matrix, I am also ready with my k 0 matrix. Now the q will be assigned initially as 0 and I will now increase the k by 1, so, k will be 1.

(Refer Slide Time: 19:19)



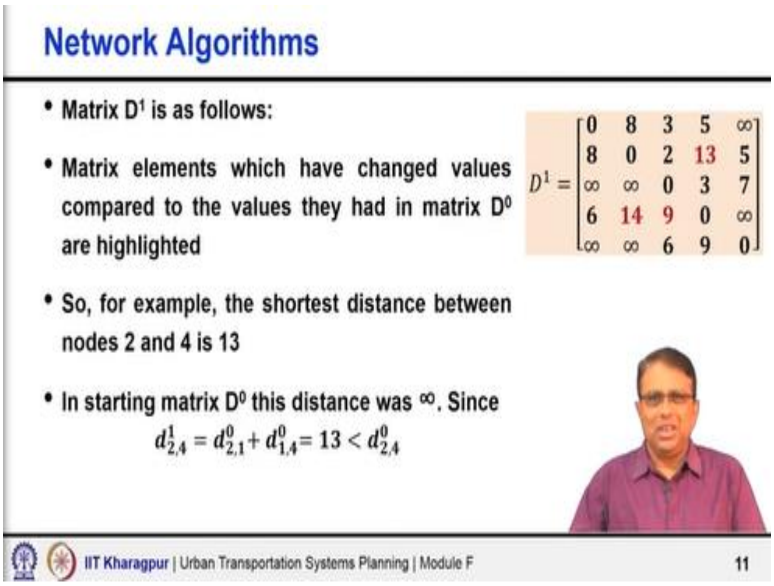
So, now what then I shall do? I shall check everything. Now in the first row, obviously nothing will change because I have already taken 1 to 2, 1 to 3, 1 to 4, 1 to 5, so, k is also 1. So, from 1 to all other home node, this will not be anything different, so, obviously myself will not change. But look at these 2 to 1, 2 to 3, 2 to 4 and I have said what are my values and I want to take the minimum.

You see when I think 2 to 4 initially what was the connection? 2 to 4 was infinity. There was no direct connection but now when I take 1 as my intermediate node, then I am exploring 2 to 1 and 1 to 4. So, when I take 2 to 1 and 1 to 4, I find the total cost is 8 + 5 is 13. 13 is less than infinity so I update my cost. Now that particular cell I shall have that and I also have to update then for 4 the predecessor node will no more be 2 to 4.

The predecessor node will not be 2 but it will be changed to 1. Same way you find d 4 2 there was no direct connection but when I travel through 1, 4 to 1 and 1 to 2 then my cost is 14, so 14 is less than infinity. So, I update my cost and I change I have changed my predecessor for node to destination node and home node 4. So, 4 to 2 if I have to travel, I know now my predecessor node for 2 should be 1 instead of 4 directly.

That I need to update 4 to 3 also there was no direct connection the cost was infinity and I find I can travel from 4 to 1 and 1 to 3 with the cost of 9. So, these are the three cells which can be or should be updated and the corresponding predecessor cell also will change.

(Refer Slide Time: 21:59)



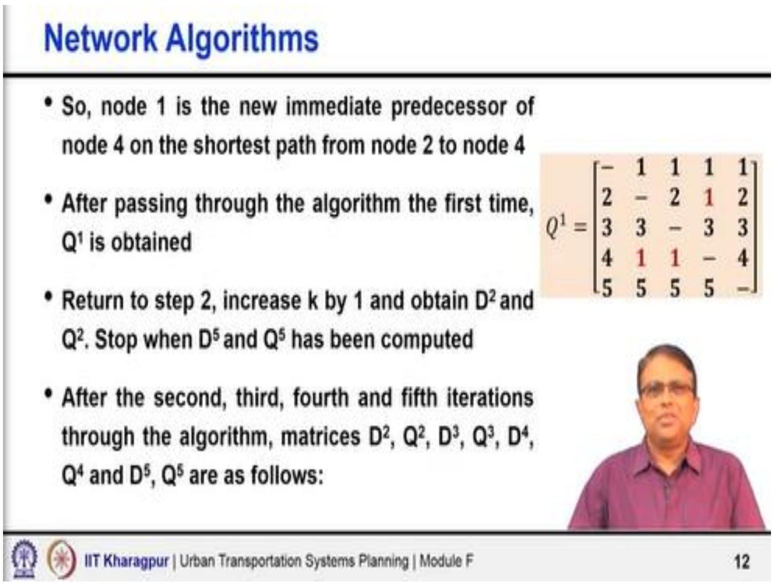
So, let us see that you can see here I have shown now the D 1 matrix and the cells which have been updated. I have marked the cells using red font red color. So, you can see 2 to 4 and 4 to 2, 4 to 3 these are the cells which were in D 0 they were all infinity they are now changed and because I got a better solution than infinity. So, I have explored but not everything we could explore.

Still you can see in this matrix there are lot many other cells which are still infinity because to those cells they neither there is any direct connection nor there is any direct connection through 1. Say for example, 4 to 5 is still infinity that means 4 to 5 no direct connection and 4 to 1 and 1 to 5 also I cannot do because at least I am sure I do not see the network here. But at least one of those links or maybe both links are there is no connection direct connection.

So, I do not have or so far nothing better in their direct connection because as we proceed this all getting updated. So, direct connection may vanish after some time, but whatever is my best solution till the previous iteration. In this iteration, we could not get a better solution than what I already got till my previous iteration. So, I do not change these cells. So, I have my D 1 matrix is available.

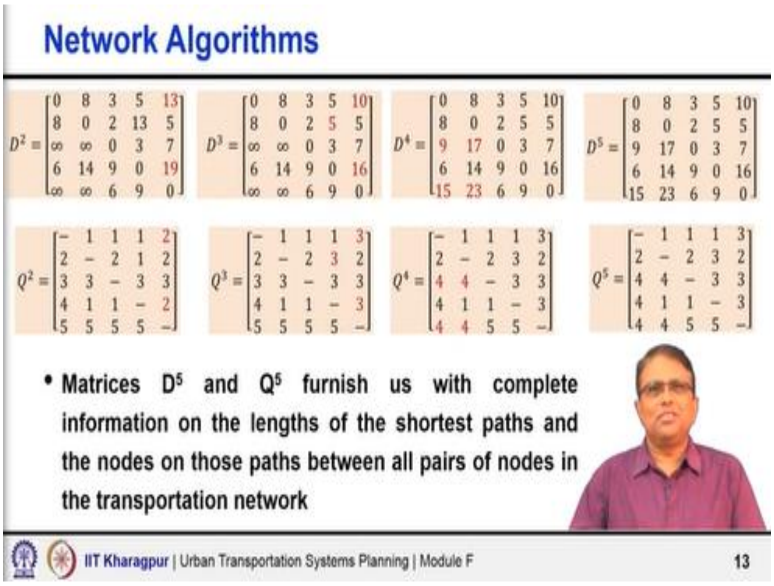
Now, the cells were the distance has changed, why the distance has changed? Because we discovered that if I travel through 1 then to all these nodes which are marked red distances as shown as red. Then I am able to reduce my cost. So, obviously I have updated.

(Refer Slide Time: 24:09)



So, now the predecessors are also updated you can see the corresponding cell the predecessor node has become 1 because by k was 1. Now, I similar way, I will now increase k by 1 more value. So, I will check each case if 1 to 3 is better or 1 3 3 2 is better if 3 2 is better 1 3 + 3 2 is better then, predecessor of 2 will be changed from the previous 1 to 2.

(Refer Slide Time: 24:40)

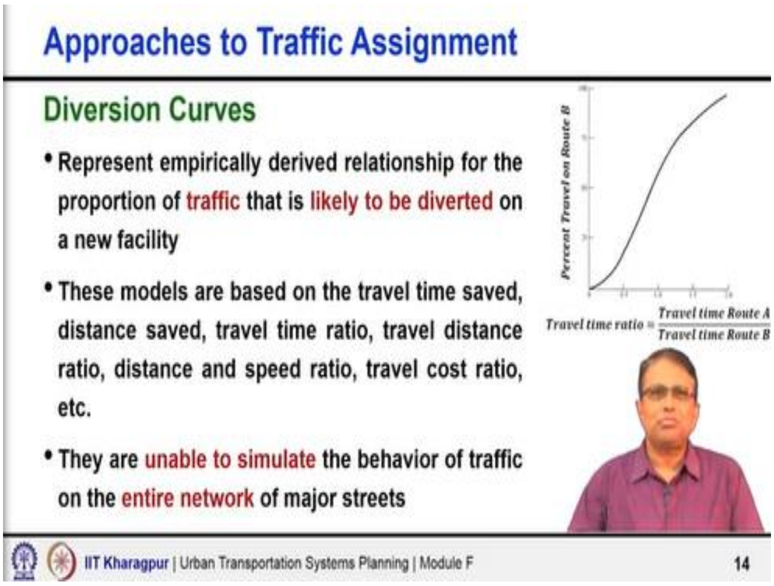


So, like that I have shown here the corresponding D 2 and Q 2 matrix D 3 and Q 3 matrix D 4 and Q 4 for matrix and D 5 and Q 5 matrix. So, each cell wherever the changes are done are also indicated using red font. So, those are the cells and you will see in all cases the corresponding predecessor cell is also changed depending on what is the Q. So, like that by the time you have reached to the nth iteration you have explored actually all possibilities.

So, whatever is my D matrix now D n all elements, whatever I got inside, they are the best value. I cannot improve anything because all possibilities I have explored. Similarly, that is what the predecessor and how to matrix? Suppose you find let us take a solution, 4 to 5 the predecessor is 3. So, I know 5 is coming from 3. Now, I shall take 4 to 3 what is the predecessor?

I find it is 1. So, I know 3 is then coming from 1. So, 5, 3, 1 and what is the predecessor for 1 is 4. So, 4, 1, 3, 4, 1, 3 and 5. That is the path and the cost is correspondingly we are talking about i 5 i 4 j 5 so corresponding value is 16, so 16 is the total cost fine.

(Refer Slide Time: 26:30)



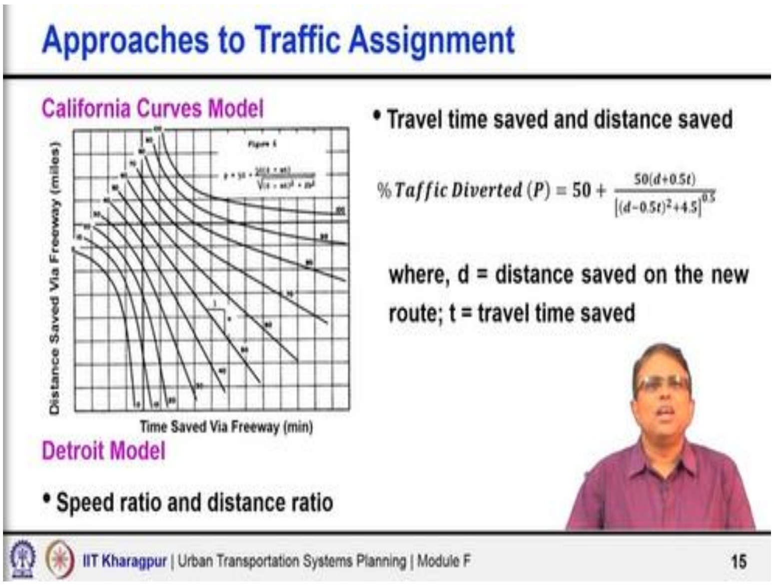
Now, another approach is also used what is called Diversion Curves. It empirically represents empirically derived relationship of the proportion of traffic that are likely to be diverted to a new facility most cases once they are constructed. So, these models are based on the consider many factors say what is the travel time safe what is the distance safe sometimes the take travel time ratio sometimes they travel distance ratio.

Sometimes take both distance and speed ratios sometimes when a cost ratio based on all this thing, they give then what is the proportion of traffic that are that will that will get shifted to the new route based on these values as I say multiple things are there. But remember they are not able to simulate the behaviour or traffic on the entire network. But mostly used when you are just considering maybe a road passing through a town and you construct a bypass route.

And you want to find out how much traffic will get diverted to the bypass. You can use simple model like the diversion curves, you know, many people many researchers have developed such kind of curves and whatever if it is based on speed ratio cost ratio, time ratio or the distance save, time save whatever is the based on what value the diversion curves are developed you can accordingly use the diversion curve to get the proportion of traffic.

That means we are not doing any formal assignment, but just getting how much grossly the traffic will get diverted based on these factors.

(Refer Slide Time: 28:15)



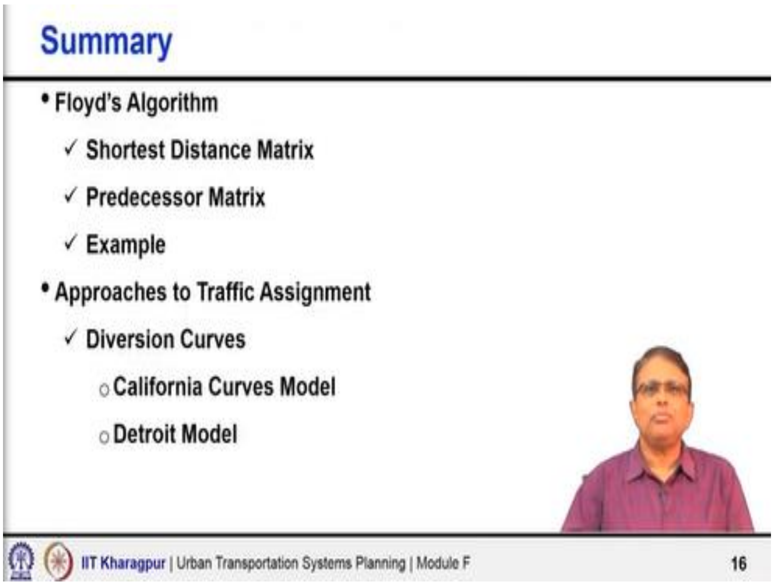
Now one such diversion curves I have mentioned here is the California curves model which actually use time saved and distance saved both. So, x axis wise you select once you are consideration if we travel by a freeway, a control access control road freeway, what is the time saved visa verse what is the distance saved? So, once you know, the time saved and the distance saved you can select one of these appropriate curves based on speed.

And then you can select an appropriate curve and then find out what is the so you know, the time saved, you know, the distance saved accordingly you can find out what is the proportion that will get diverted. Now some people may consider even a third for parameter forth parameter accordingly various representations are there. So, basically these are the aspects which are considered some may consider only based on travel time ratio.

Some may consider cost saving, time savings, distance saving, distance ratio many factors people consider. And based on those once you know, my proposed facility values my new existing or the current facility value then what is the saving or what is the ratio in that whatever factor is considered then from the graph you can say, my 20% traffic will get diverted.

Similarly, Detroit Model is also another model which takes speed ratio and distance ratio. So, as it takes it took travel time savings and distance saving. So, similarly the Detroit model say speed ratio and distance ratio. So, if you know the speed ratio if you know the distance ratio you can find out how much you know, from the diversion curves you can find out how much will be the shift of the traffic to the new route or proposed route.

(Refer Slide Time: 30:26)



So, in this lecture we discussed about the Floyd's Algorithm. I explained you how the algorithm works and also try to explain you with an example problem step by step how you get. Then also introduced to you the approach another approach for traffic assignment which is diversion curves and what is diversion curve? What are the kinds of parameters we use to estimate the diverter traffic or the proportion of traffic which are likely to get diverted?

And two particular models I mentioned one is the California Curves Model and another is the Detroit model. So, with this I close this lecture and thank you so much.