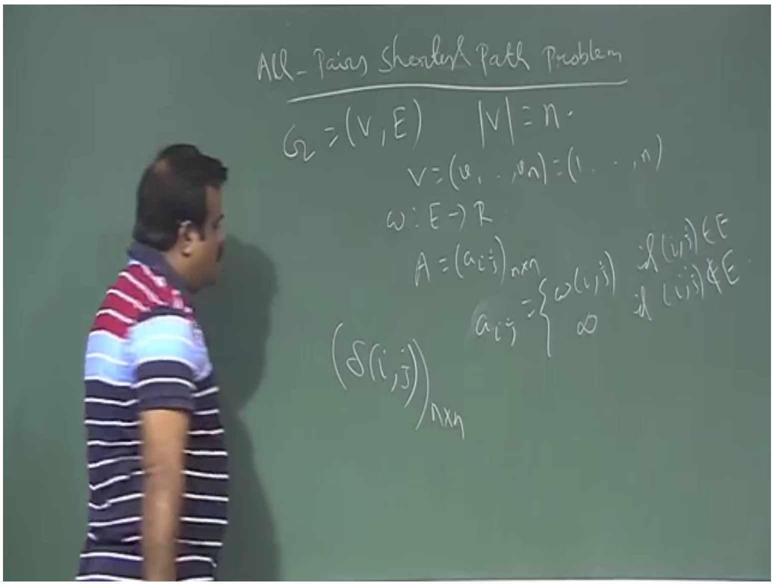
An Introduction to Algorithms
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 49
Floyd – Warshall

So we are talking about all pairs shortest path problem. So, we will we will discuss the today we will discuss the Floyd-Warshall algorithm. So, let us just recap. So, we are talking about all pair shortest path problem.

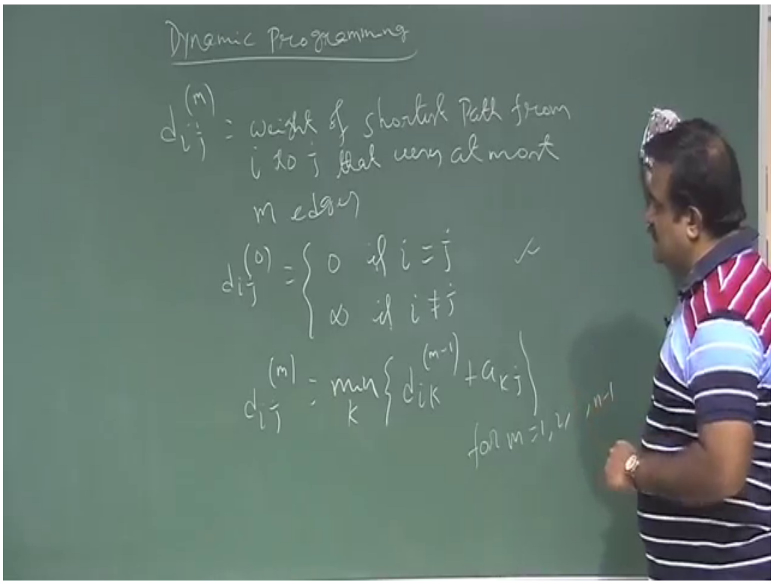
(Refer Slide Time: 00:30)



So that means, given a graph v comma e directed graph we have this v is n. So that means, v is beyond u to v n and for simplicity we are taking 1 to n. And we have the edge weight t w which is e to r and which you are taking in adjacency matrix A I z. So, A i j is basically this is n cross n matrix this is the adjacency matrix. So, A i j is basically w i j if i j is an edge otherwise it is infinity if i j is not is a edge you cannot put it 0 because 0 could be the weight of the edges ok.

So, that is the danger with putting 0. So, this is the these are the input and what we have to find out you have to find out the all pair shortest path you have to find out delta of i j. So, this matrix So, delta is the weight of the shortest path from i to j. So, for this we are talking about dynamic programming technique. So, in the last class we defined some D i j's.

(Refer Slide Time: 01:53)

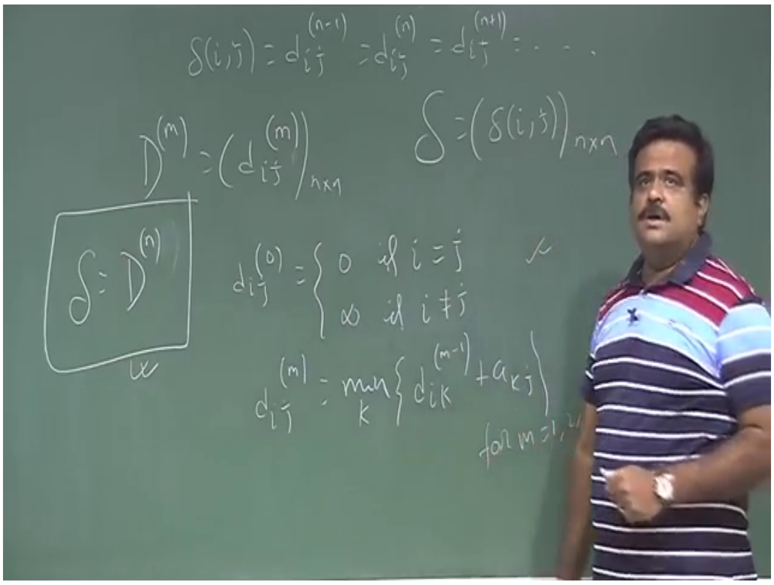


So, you have defined this D i j m this is basically weight of the shortest path from i to j which uses that uses at most m edges ok.

So, that is the weight of the shortest path that uses at most m edges. So that means, if we I from j if you go and if we just count on the weight if we just count the shortest if we just count the number edges in this path, then that number must be less than equal to n. And we take the minimum weight such weight So that is our D i j m. And we in the last class we have seen this theorem. So, so it is base case D i j 0 we know it is 0 if i is equal to j, otherwise it is infinity if I not equal to j and you have seen this for i j m is basically minimum amount all k such that D i k m minus 1 plus a k j. And this k is varying from 1 to n.

So, this is for this is for all m 1 to up to m minus 1 and for m is equal to 0 this is the expression. So, this result we know from the last class we prove this in the last class. So now, we will use this result to have the delta. So, what is the relationship between deltas on dijs? That also you have discussed in the last class. So, basally if there is no shortest path I sorry, if there is no negative cycle, then there will be shortest path in that case delta i j is basically D i j to the power n minus 1.

(Refer Slide Time: 04:21)

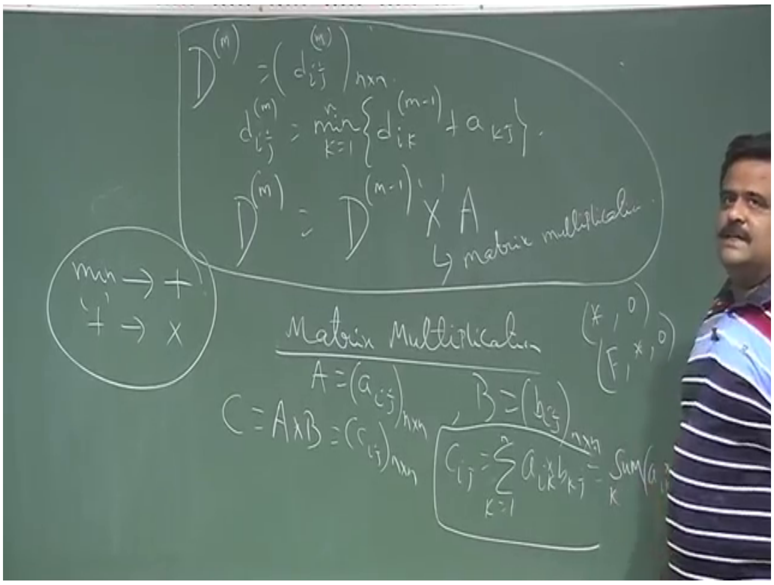


Because, if there is no negative cycle then the shortest path will be simple path so that means, we have at most m minus 1 edges in the in a path.

So, which is basically converge this will be basically delta of m that is why D i j m plus 1 and so on. So, if we write this in a matrix form if we write D matrix D to the power m is basically this D i j to the power m this n cross n. So, if you take this D i j in i j th entry of this matrix, then basically delta which is the delta is a matrix. So, delta is basically D i j delta matrix is D n minus 1 or D n we can say.

So, basically all pairs shortest path problem is to find this delta matrix capital delta. So now, the question is how you can get this matrix, how you can we get D to the power m or D to the power m minus 1 or D to the power n in general because it will converging there is no negative weight edges. So, let us try to get D to the power n. So, that will give shall being us to finding the all pair shortest path. So, our goal is to find D to the power n.

(Refer Slide Time: 06:02)



So, where D to the power n is basically D i j to the power n ok.

So, what is So, D to the power n is basically what D to the power n is D i j to the power n. So, how D i j is basically minimum over k of this expression. D i j to the power m minus 1 plus a k j. Now we will we will we will just how to get this D matrix. So, for that you need to take help of what is called matrix multiplication. So, suppose we will related this expression with matrix multiplication. So, we want to claim that this is basically b n is basically D of n minus 1 into a matrix. This is sort of some sense matrix multiplication. So, what is that? How we can view that in this way? We will discuss. So, how to defined this multiplication? How to defined this what is the operation for that? Let us just go back to the what do we mean by matrix multiplication of 2 matrix say real matrix. Suppose we have these we will come back to this. So, so just concentrate on matrix multiplication. So, suppose we have 2 real matrix a b. So, this A is nothing to do with this a.

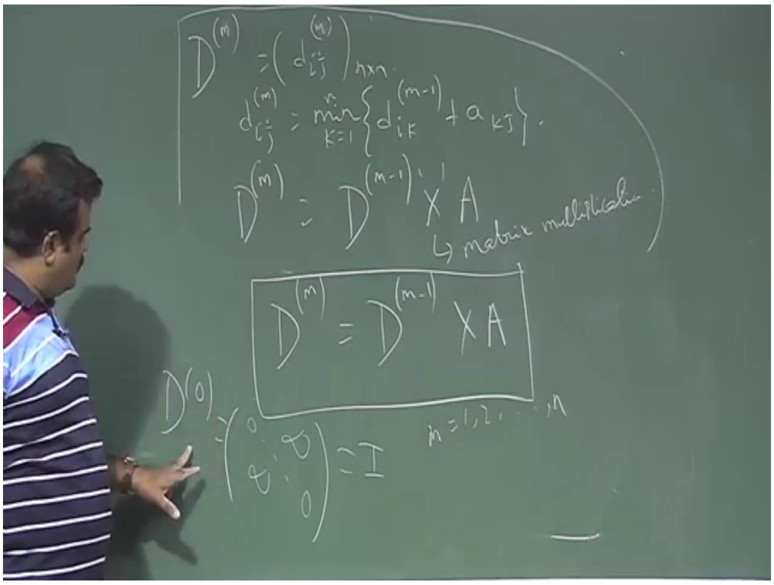
So, just keep this separate now you are talking about just matrix multiplication recap matrix multiplication. So, we are just. So, for matrix multiplication what you have you have 2 matrix suppose real numbers of size n by m it could n by n, n by n then by n p bar for the simplicity you have 2 matrix A and B both are of same size n cross n. Now then what is the multiplication of these 2 matrix? C is the A into B this is the matrix multiplication if these are real number. Or it could be over a field we will talk about that. So, then what is the c, c is also basically c i j which is also size n by n. And c i j is basically summation over A i k v k j and this k is from 1 to n this is the formula for matrix multiplication.

So, this is this is into this is into So, this into is the real number. So, if A I n these are real number this is the real number multiplication 2 real number multiplication and this is the real number sum. So, these we can write as this way. So, just we can write it as basically if we have a function sum, sum of a k over k then A i k into b k j that way you can write. So, can you relate this expression with this expression. So, we can just the instead of some operation if you think this is the minimum operation. So, minimum among this all case. This k is varying from 1 to n. And then addition if we replace by this multiplication this real number multiplication. Then this is nothing but a matrix multiplication. Just think about it. If we just relate this expression with this expression, this minimum is basically the this sum. And this multiplication is basically this addition.

Then this is basically a matrix multiplication. So, if just relate this if we just this minimum as a sum and then this class as a that that into then this is nothing but the same expression as this. So, in general if this numbers are coming from a field this we are talking about real field, if in general it is coming from a general field and we have a operation on the general field on operation means multiplication operation another operation is summation operation this summation. So, similarly if we have another if we have a operation say star and say this symbol then also we can defined this matrix multiplication over this field suppose we have a field with this 2 symbols. Provided this satisfies some of the properties it has to be filled. So, provided this minimum and drop last we will satisfy the those field properties. So, let us assume these a satisfying those properties we can one can prove this.

So, this is nothing but a matrix multiplication, if you compare this expression with this. So that means, this is our a matrix basically. So, A is basically this is coming from D n minus 1 and this is our matrix which is basically compare to b matrix. So, this is basically the adjacency matrix. So, that is it. So, this is the basically the matrix multiplication. So, he prove this. So, this is basically D i j n is basically. So, this result is D n is basically D n minus 1 into A.

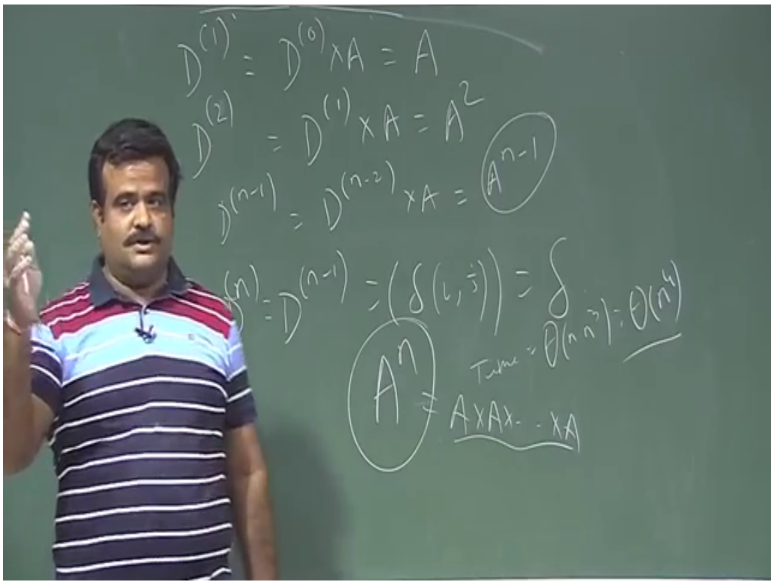
(Refer Slide Time: 12:30)



And this n is varying from 1 to up to n. And what is if we put n is equal to 1 what is D 0? D 0 we know D 0 is basically the this matrix.

So, this diagonal element are all 0 and off diagonal all are infinity. So, this is serve as A identity matrix, this serve as A identity matrix we check that under this 2 operations. Our operation is minimum and this class. So, you have to little carefully while we finding these dijs D matrix. So, our operation is minimum in this class. So, this is a good news. So, you have this dijs with this we have the expression for D m with this identity matrix. So we put this over here. So, we will use this. So, what is then D 2?

(Refer Slide Time: 13:38)

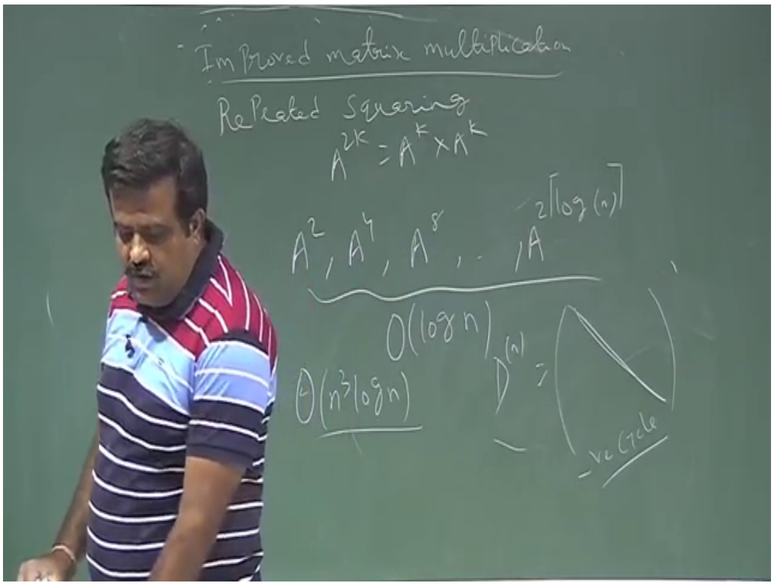


So, D 1 is basically D 0 into a now this will serve as identity matrix this is basically A. This is basically A and what is D 1 D 1 sorry D 2 D 2 is basically D 1 into a now D 1 is A, since it is basically A square. So, these way we can just proceed D n minus 1 is basically D to the power n minus 2 into a and which is basically y to the power n minus 1 ok.

Now So that is So, this will give us. So, A is the adjacency matrix. So, basically now if we can find D to the power n minus 1 that will give us basically delta of i j delta matrix. So, to find D to the power this we have to find or which is same as this is converging. So, which is same as if there is no negative cycle. So, basically the problem is to find A to the power n say, if we can find A to the power n then you are done. So now, the question is how to find A to the power n.

So, this is basically matrix multiplication. So, this is a how many times matrix n times. So, each time matrix multiplication will take this is with matrix of size n cross n. So, order of n cubes. So, time will be n into n cube. So, n to the power 4 which is not as good as if you run the bellman ford for all your shortest path. So, how to simplify this? We can use this we can simplify this matrix multiplication little bit in this is the nave approach. So, this is what this is A in to A into A. So, n times. So, each time we are doing a matrix multiplication with a cost order of n cube A provided we can handle that 2 operation minimum or class. Because your not in real matrix. So, you have to little careful about the field. So, our field is just this operations are minimum operation and the plus operations. So now, how to improve this matrix multiplication? So, you can improve by just one can improve by this repeated squaring.

(Refer Slide Time: 16:21)

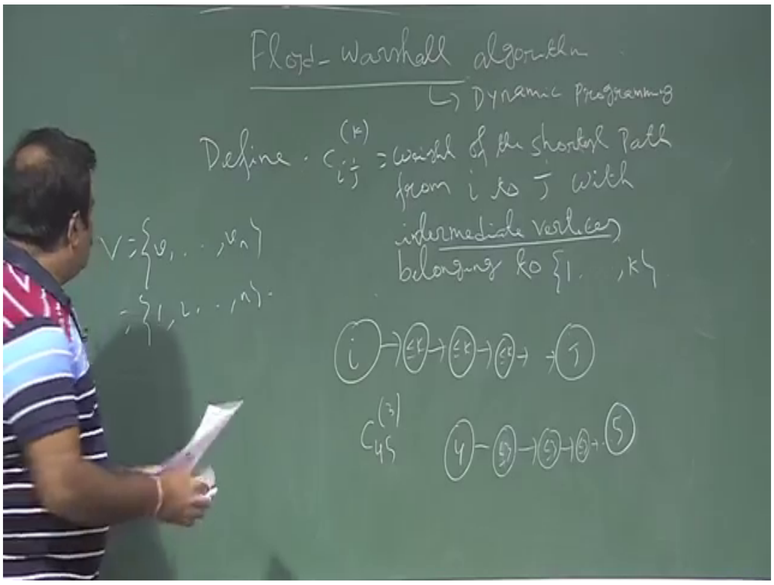


This is the improvement of improved matrix multiplication. So, you can improve by repeated squaring.

So, how we A to the power 2 k is basically A to the power k into A to the power k. So that means So, we have to find A to the power n. So, to find A to the power n what we can do we can a find A square A 4 then A to the power 8 like A to the power 2 into log of n. Sort of lower ceiling or upper ceiling any way. So, this is the way. So, this will give us how many time this is log on log n such squaring. So that means, it will take order of n cube log n which is not benefit from that because, yeah earlier it was order of n to the power 4 now it is order of n cube if we do this multiplication little carefully and to detect the negative cycle. So, if we have to detect the negative cycle then it will not converge. And we have to take we have to check the diagonal element. And so, we find D to the power n say, and we check the diagonal element. And if the diagonal element has negative value then there is a negative cycle. So, this is how we detect the negative cycle ok.

So now we will discuss a better algorithm which is called Floyd-Warshall algorithm which will give us the order of n cube time for finding the these deltas. So, for that we defined new variable c i j to the power k. So, let us just Write this Floyd-Warshall algorithm.

(Refer Slide Time: 18:36)



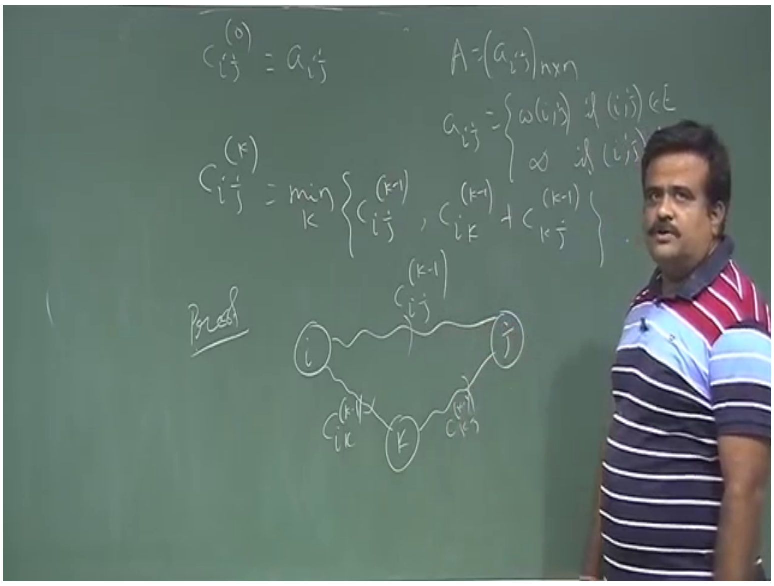
This is also a dynamic programming algorithm, but which is faster than the one you have seen. So, Floyd-Warshall algorithm. So, this is also dynamic programming technique ok.

So, what we are doing here? We are defining. So, instead of D i j here we are defining c i j to the power k. So, this is basically weight of the shortest path. So, here we are numbering the vertices we already have numbered the vertices v 1 v 2 v n. So, this is basically 1 to up to n for the simplicity. So, weight of the shortest path weight of the shortest path from i to j node i to node j with intermediate vertices values are less than equal to k, with intermediate vertices belonging to belonging to the set 1 to k. So, what is the meaning of this? Suppose you are at I you want to go to j and in the middle he visit some vertices dot, dot, dot. So, so these vertex numbering must be less than equal to k that is the meaning. And you have no restriction on i and j, i and j could be more than k, but you have a restriction on the intermediate vertices.

So, the intermediate vertices we see they are their number must be less than less than equal to k. So, that is how we define c i j k. So, for example, if we just find out c 4 5 3. So, this is basically we have our vertex 4 vertex 5. Now we consider all the path from 4 to 5 such that if we see a vertex over here, intermediate all the intermediate vertices dot dot their either one there must be less than equal to 3. So, they are either v 1 or v 2 any of these either v 1 or v 2 or v 3. So, this is the restriction all the vertices must be less than equal to 3. And we take all such path and we take minimum among that. So, this is how we defined this cijs.

Now, let us have the recurrence following c i j. So, that we can have the manual programming technique for that.

(Refer Slide Time: 22:03)



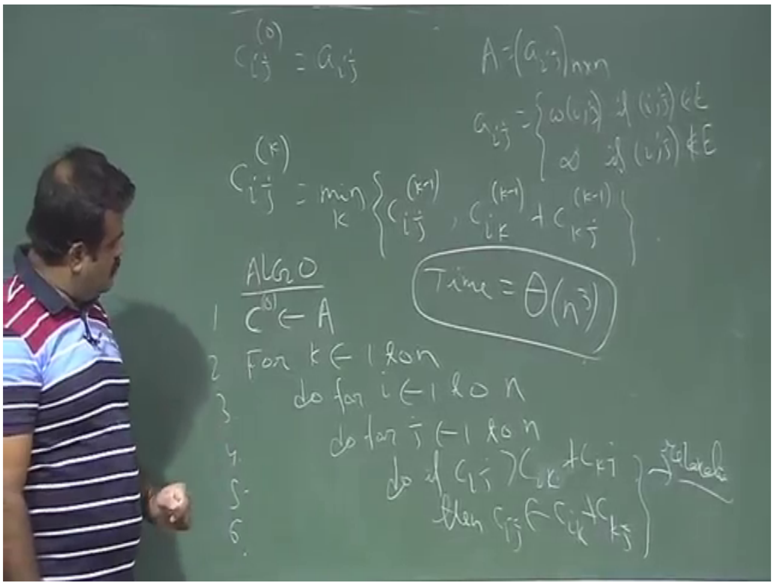
So, so what is c i j 0? What is the meaning of c i j 0? C i j means we are at i th vertex we want to go to j th vertex. So, we because there is no vertex number s v if we have vertex in the middle there has to some numbers. So, you are not allowing that. So that means, there is direct edge. So, this is basically A i j. A i j is the adjacency matrix A i j is the n cross n where A i j is w i j if i j is an h otherwise it is infinity if i j is not a reason h ok.

So, c i j 0 is A i j, and what is the other expression for c i j? So, c i j k we have to write in terms of k minus 1. So, we claim that c i j k is basically minimum amount c i j k minus 1; that means, we are not seeing the vertex k th vertex, I mean vertex which is number k or c i k k minus 1 plus c k j k minus 1. So, how to prove this. So, this k is from how to prove this. So, we are at i th vertex we want to go to j th vertex ok.

Now, it may happen and we do not want to see a vertex whose number is more than k. So, it may happen that we are not seeing at all the k th vertex. So, that is one possibility that is c i j k minus 1. And the another possibilities is we must see the k th vertex in the middle. So, if you see the k th vertex then we go to the k th vertex then this. So, this is basically c i k k minus 1 because in the middle we do not want to see k th vertex because if you see k th k th vertex that is a self loop that is the cycle. And since shortest path will exist if there is no negative cycle. So, they we can ignored that cycle.

So, similarly this one is also c k j k minus 1. So, these are 2 possibilities. So, minimum among this 2. So, this is a path we take this plus this and these. So, these 2 are possibilities we take minimum among these 2. So, this will give us this is the proof now. So now, how this c i j k will help us to have a delta. So, basically we need to find out the delta. Delta is A So, what is the relationship between delta and c i j?

(Refer Slide Time: 25:05)



So, basically delta of i j Which is basically shortest path from i to j is basically c i j to the power n, why? Because if there is no negative cycle. So, every path will contain simple every path is simple path.

So, you are going to i to j. So, the path you visit there is the maximum i vertex are visiting is up to the vertex numbering is up to n. So, that is basically delta of i comma j. So, if we can find c i j n we are through. So, these will be we are getting from this recursive algorithm. So, what is the algorithm. So, basically we need to find c i j with the initialize by this. So, let us have the algo Floyd-Warshall algo.

So, this is basically we just initialize by c matrix as a this is the c i j 0 then for k is equal to 1 to n and do for i is equal to 1 n and do for j is equal to 1 n, what we do? Do if c i j is greater than c k j sorry, c i k plus c k j then you must relax this c i j. Then c i j will be relax c i k plus c k j. So, this is sort of relaxation we are doing. So, this is 3 4 5 6. So, that is it.

So, this is basically your calculating this c i j recursively. So, this is initialized by a matrix this c if we denote this c matrix then. Then after that this c i j is c i k is doing like this. So, this is basically c I 0 and then if we want to have this index otherwise this is they, because we are taking the variable then we are updating their itself. So, this is the code, if this is greater than we are just relaxing, otherwise this will be the earlier one. This is the earlier one c i j in k minus 1. So, what you is the time complexity for this? Is basically we have how many loops you have 3 for loops. So, time is basically order of n cube which is better than the earlier one. So, this is what is called Floyd-Warshall algorithm. So, in the next class we will see some application of Floyd-Warshall algorithm.

So, so this is the better algorithm than So, this is also a dynamic programming technique.

Thank you.