INDIAN INSTITUTE OF TECHNOLOGY MADRAS

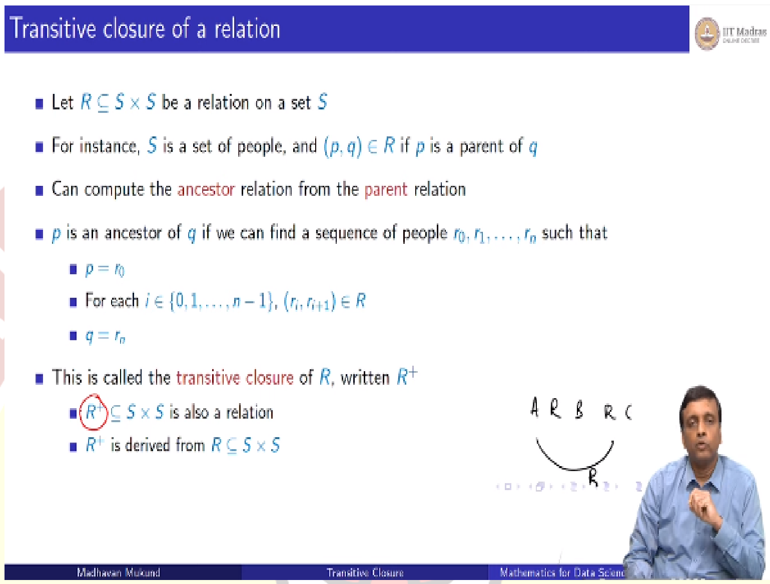
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

Mathematics for Data Science 1
Professor Madhavan Mukund
Indian Institute of Technology, Madras
Lecture 70
Transitive Closure

(Refer Slide Time: 0:18)



One of our original motivations for looking at graphs was to visualize relations. So, let us go back to relations. So, supposing we have a relation R on a set S. So, relation R on a set S remember is a subset of the Cartesian project S cross S, so S cross S is all pairs S1, S2 taken from S and some subset of these form a relation.

So, concretely, for instance, supposing we have a set of people, maybe a family, and then we want to find out the family tree in some sense, then we might represent as a relation when two people are related as parent and child. So, we can say that p, q belongs to R so, R is the parent relation, whenever p is the parent of q, so, p, q belongs to the relation R, if p is a parent of q.

Now, given this parent relation a very natural question is, what is the grandparent relation, what is the great grandparent relation and so on. So, in general what is the ancestor relation. So, we have is so and so is p an ancestor of q, is in the family tree is q a descendant of p? So, how would we do this? Well, to find out whether q is a descendant of p or p is an ancestor of q, we have to trace some sequence of relationship, we have to find a child of p and for that child, we have to find child of that child and so on or we have to find a parent of p.

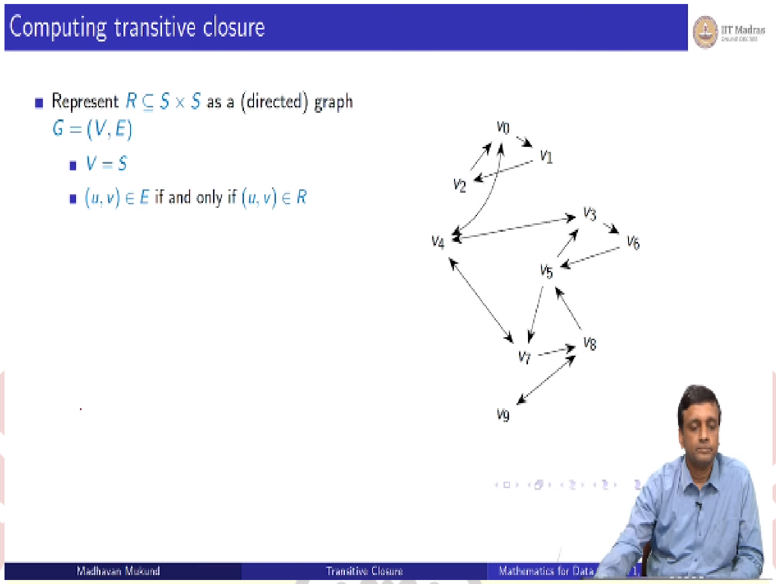
So, in this case, we are looking for ancestor so, p to q is a parent child relationship. So, we start with some p let us call it R_0, we have to find the sequence R_0, R_1, R_n we have to find the sequence of people says that R_0 is a parent of R_1, R_1 is a parent of R_2, R_2 is a parent of R_3 and so on, R_n-1 is a parent of Rn, so this is an ancestor sequence. So, R_0 is an ancestor of everybody to the right in sequence and particular if R_0 is P and R_n is qr, these are our two people we are interested in finding out, then we have established that p is an ancestor of q.

So, this is a new relation. So, this relates pairs of people who are connected by a sequence of parent relations. So, this has a name, this is called the transitive closure. So, transitivity says that if A is related to B and B is related to C, then A must be also related to C, this is the definition of a transitive relation.

So, parent is not a transitive relation, so what would happen if we made parent a transitive relation, if we forced parent to be a transitive relation, so, we said that whenever somebody is related to somebody and somebody is related to the other person, first person is related to third person, then that is the ancestor relation.

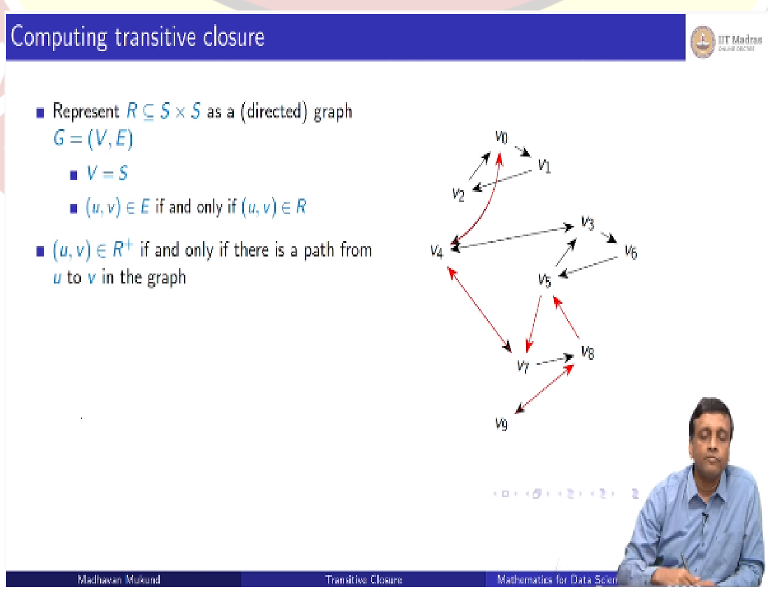
So, this is what happens when you close, we make the parent relation closed under transitivity, we force transitivity onto it, we compute what is called a transitive closure, okay, this is what we have. So, we normally denote this by R^+, R^+ means that we have to apply R one or more times to go from p to q. So this is a transitive closure of R.

(Refer Slide Time: 3:32)



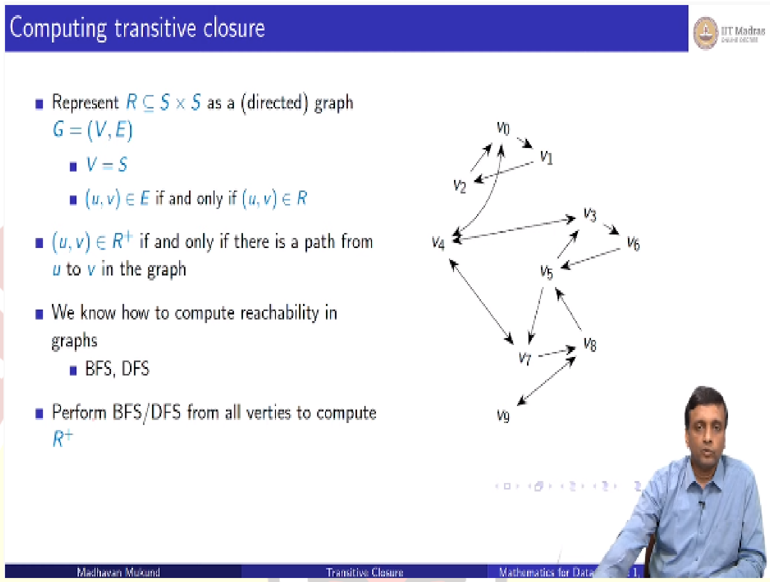
So, our question is, of course, how are we going to calculate this transitive closure in a systematic way? So, recall that we can represent any such relation as a directed graph in general. So, we represent, the vertices represent the set S, V=S and each edge represents a pair in the relation. So, if u,V is a member of the relation R, then we put an edge from u to V and this is the only case in which you put an edge from u to V. So, the edges are exactly the pairs which are related by R.

(Refer Slide Time: 4:07)



So, what we have defined as R^+ can be calculated in the graph as being a path, right, so we can say in this case that V9 is related by R^+ to V_0, because there is a path from V_9 to V_0.

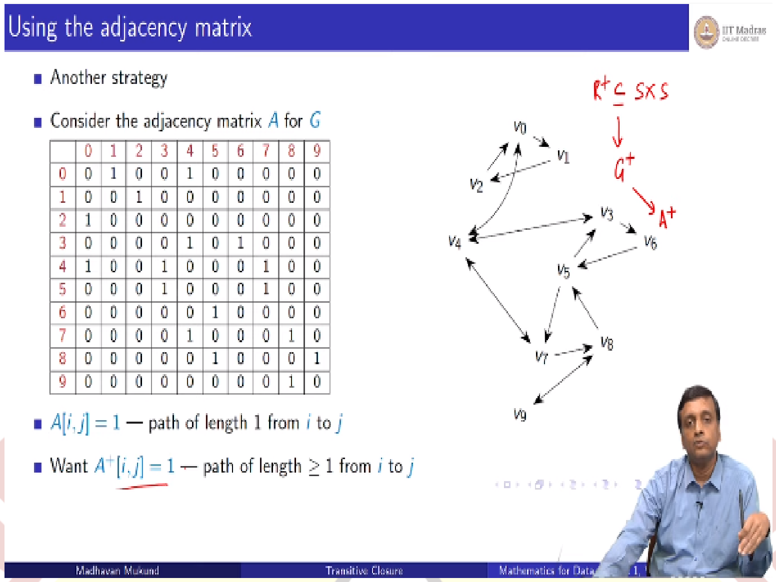
(Refer Slide Time: 4:25)



So, how do we find these paths? Well, essentially, we want to find all such pairs for which V_9 is related and we said that if you do not focus on a single path, we can just calculate reachability, what all is reachable from every vertex and this we can do using breadth first search and depth first search. So, in the R^+ case, we are interested in finding the reachability for every i in every j, right, we want to know for every i and every j, whether it falls into R^+ or not. So, we have to compute this reachability for every starting point.

So, one way to do this is to just perform this BFS, DFS starting systematically we started the 0th vertex, you perform BFS, then you know everything which is of the form 0, i, then you start at 1 and you perform, notice that this is a directed graph. So, you have to do it because if there is a, if 0 can reach j, it is not obvious or is not required that j can read 0 in a directed graph. So, you have to perform this from every vertex to find out what all pairs fall into R^+. So, this is one way to do this using what we already know.

(Refer Slide Time: 5:25)



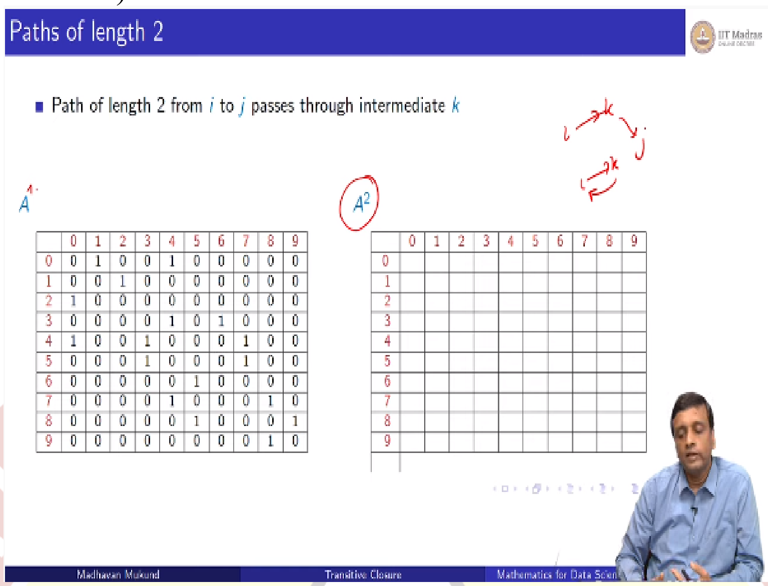
So, here is another strategy, so the other strategy is to look at the adjacency matrix. So, in this adjacency matrix, we put a 1 if there is an edge from i to j. But another way of thinking of an edge is that there is a path of length 1 from i to j. A single edge is a path of length 1 and what we want is to create an similar adjacency matrix for this expanded list.

So, remember that R^+ is also relation, so R^+ we said is also a relation, so this also correspond to some new graph G plus, because every relation we can draw a graph, just add the same set of vertices in this case, but now put an edge if there is an R+ relation value, relationship between i and j.

So, this G^+ will correspond to an adjacency matrix A plus, so, that is this A+ here, we want an matrix A+ whose ij entry is 1 , if there is a path of length 1 or more from i to j, so, if there is an edge directly, so an ancestor, a parent is an ancestor, parents parent is an ancestor path of length 2, parents, parents parent is an ancestor, so that is length of 3 and so on.

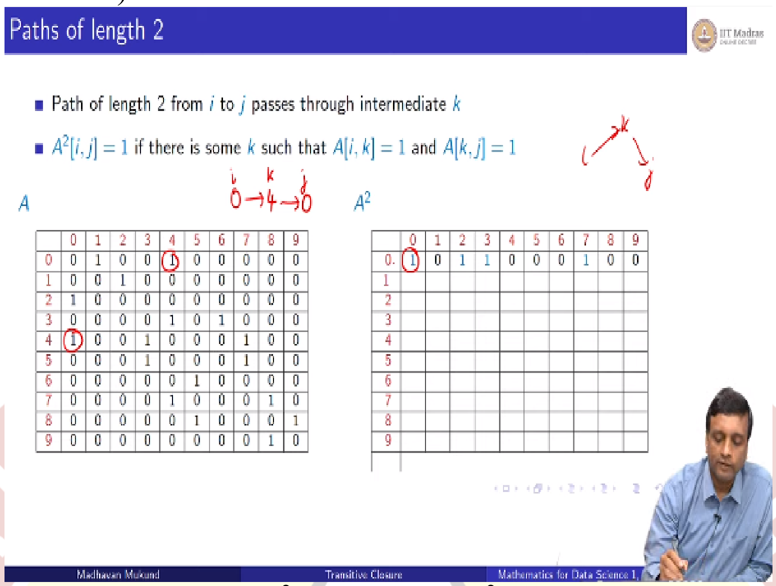
So, therefore we have any path of length 1 or more between i and j that will be an edge in the R^+ relation and in the G^+ graph, and we want to compute A^+ from A, so that is our goal. So, we know one way to do it, which is to go to breadth first search, and then fill in A+ from breadth first search by doing breadth first search from every vertex. But the question is, can we do it directly using just A?

(Refer Slide Time: 7:08)



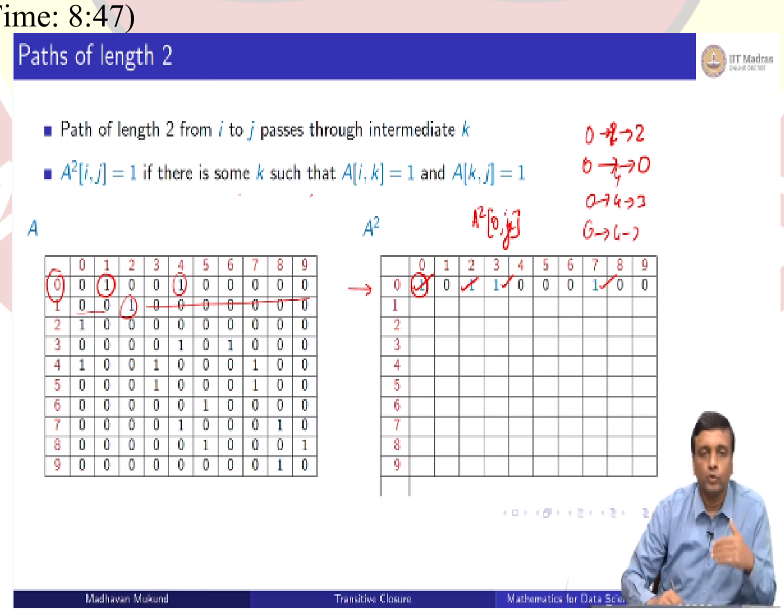
So, we have now on the left, we have A and A denotes paths of length 1. So, as the first step let us try to compute paths of length two, right. So, paths of length 2 go from i to some k, to some j. So, this is a path of length 2, we could also be back to itself, path of length 1 cannot go but we could have a path of length 2, which goes from i to i. So, this is the matrix that we want to call now A^2 for instance, A^2 so A, which we can think of A^1 if you want, represents paths of length 1, A^2 represents paths of length 2.

(Refer Slide Time: 7:55)



So, how do we fill in the entries of A^2 from A? So, A^2 ij is 1 if there is some k such that A_{ik} is 1. So, there is i to k there is a path of length 1 and k to j there is a path of length 1. So, I look at this entry, why is there an A^2 entry from 0 to 0? Well, I claim that there is a 0, 1 entry, not a 0, 1 entry sorry, there is a 0, 4 entry, so I have a path from 0 to 4, because of length 1, and then I have a 4, 0 entry. So, if I choose k to be 4 and i to be 0 and j to be 0, then I get A^2 ij as 1 because for k =4 I have this.

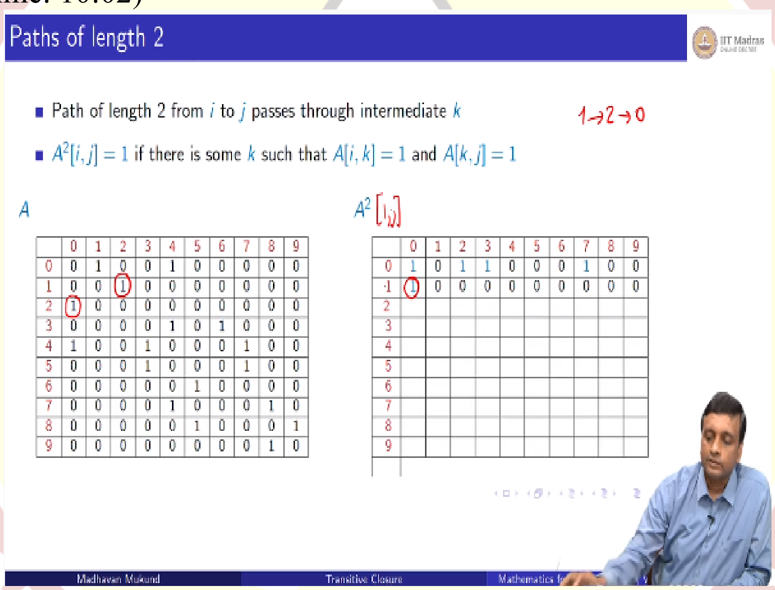
(Refer Slide Time: 8:47)



So, in general, what I have to do is I have to look at, so if I do it systematically, I will start, I want to fill in the row for 0. So, I look at all this, I look at the first entry, I can go, where can I go via 1? So, if I say 0 to 1, then I look at the 1th at the entry, so 1 can go to 2, so therefore I have an entry 0, 2. So 0 to 1 to 2, then 0 can go to 4, so I should have. Sorry, yes, so 0 to 1 cannot go anywhere else, so 1 has only one outgoing edge so through 1 I cannot go anywhere else. So, 0 can go to 4, so I now look at the outgoing edges from 4, so 0 to 4 to 0, right, so that is how I get it.

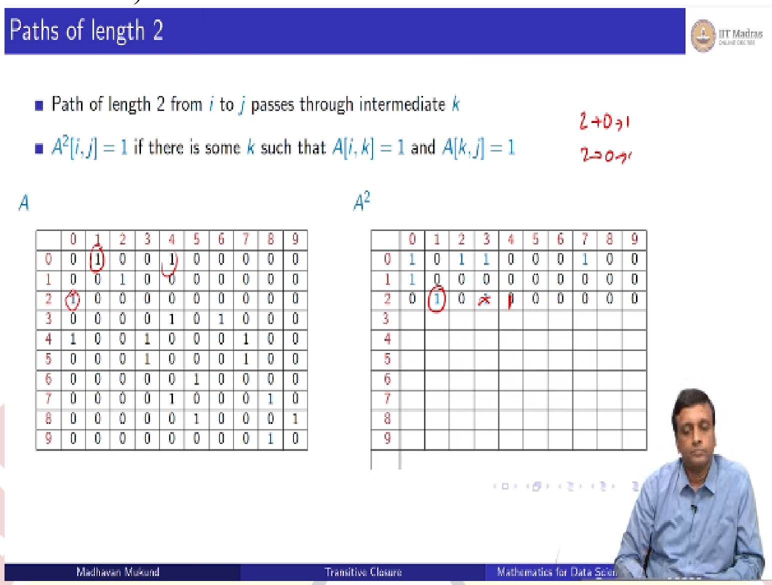
Then I have 4 goes to 3, so I have 0 to 4 to 3, so I get this entry and finally I have 4 goes to 7 so I have 0 to 4 to 7, so I get this entry. So, in this way, I can compute all the entries of the form A^2 0,k, by finding the intermediate values by looking for something that 0 goes, 0 to k and then from k to j.

(Refer Slide Time: 10:02)



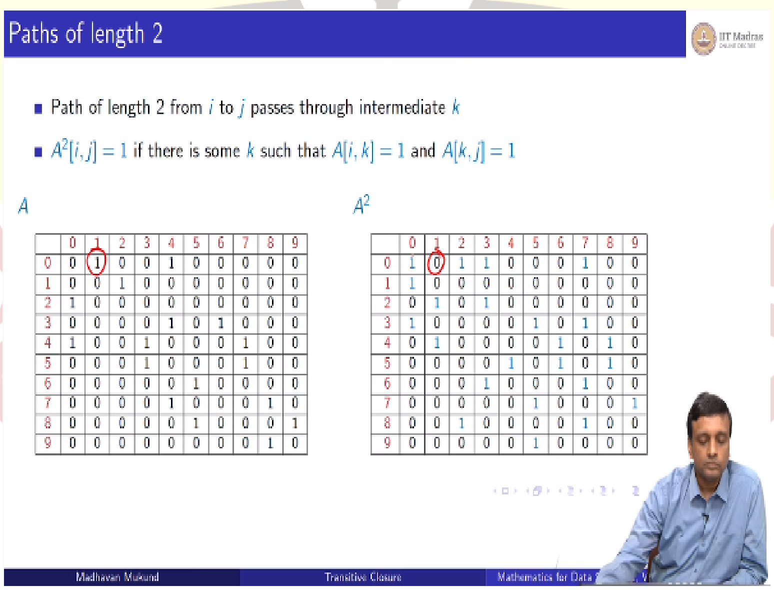
So, I can do the same thing for 1 now. So, I want to find all the entries of the form 1,j, so I look at the, so 1 has only one outgoing thing, going to 2, and 2 has only one outgoing thing going to 0, so the only new thing I discovered is 1 to 2 to 0, right, and that is the same.

(Refer Slide Time: 10:25)



Now, I will do the same thing for 2, so 2 has only one outgoing thing to 0. But 0 has outgoing edges to 1 and to 4. So, I get 2 to 0 to 1 and 2 to 0 to 4. So, I get 1, so this should not be here, this will be here, I get 2 to 1 and 2 to 4.

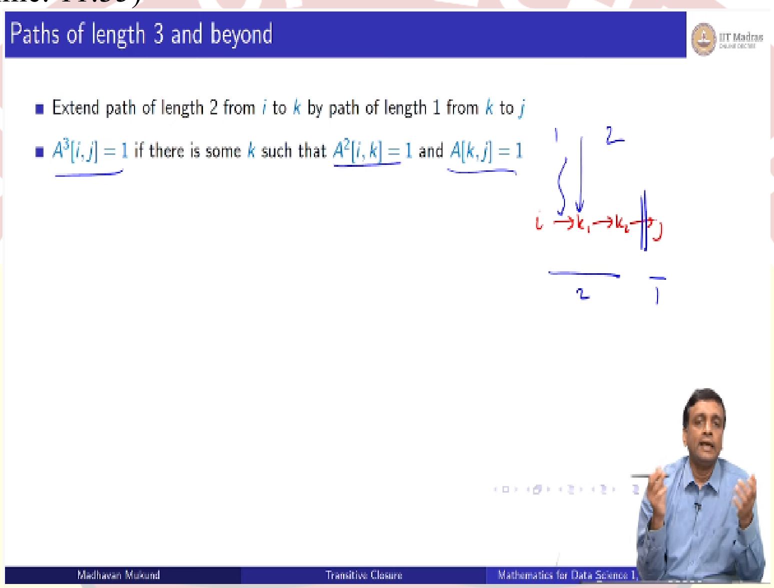
(Refer Slide Time: 10:47)



So, in this way, I can do this for all the entries, I can for every row, I can take the outgoing edges 3 to k and the outgoing edges k to j, and add an entry 3 to j, right. So, by scanning these two rows in this matrix I can compute A^2 matrix. So, A^2 represents all the paths of length 2.

So, notice that the paths of length 2 do not subsume the paths of length 1, right, so for instance, we had a path of length 1 from 0 to 1, but we have no path of length 2 from 0 to 1. So, these are paths strictly of length 2, they are not of length 0 or not of length 1, so A, the first A has edges, the second one has paths of length 2, so I can go in length 2 from 0 to 0. But the fact that I can go from 0 to 1 in length 1 does not mean I can go from 0 to 1 length two. So that is paths of length 2.

(Refer Slide Time: 11:35)

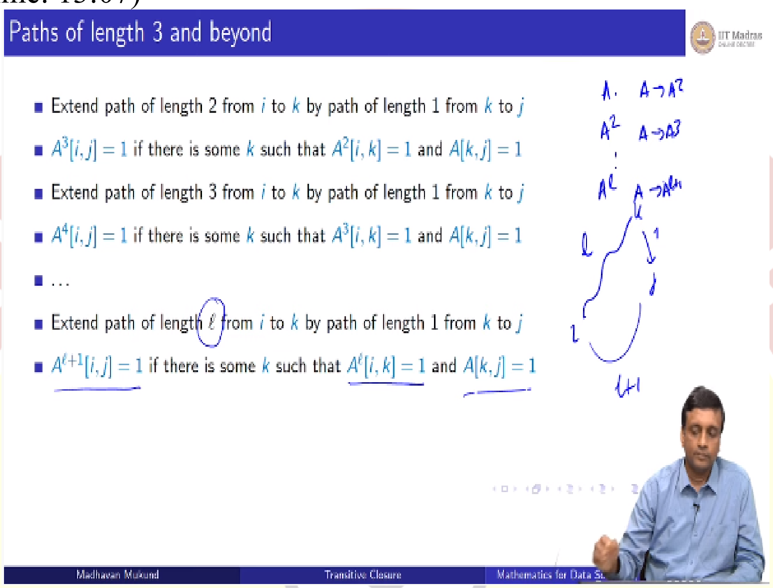


So now, how do we go to path of length 3? Well, if I have a path of length 3, it must be of this form, i, k_1, k_2 to j, right, so there must be some two things in between. So, I can split it up whichever way I want, I can either take this point and say that I have a path of length 2, followed by a path of length 1, or if I want, I could do it the other way, which is I could, I could split it up here and say I have a path at this point.

Sorry, and say I have a path of length 1 followed by path of length 2, right. So basically, a path of length 3 can be decomposed as two+ one or one+ two. And I already have explicit matrices for 2 and 1, I know all the paths of length 1 are represented in A, I know all the paths of length 2 are represented in A^2.

So, I can say now A^3, i,j is 1 if there is some k for instance, where there is a path of length 2 from A to k and there is a path of length 1 from k to j. So, now earlier, I looked at A and within A I look for two entries. Now, I look at an entry in A^2 and I look for an entry in A and I try to match them up, I tried to find a k such that from i to k have an entry 1 in the A^2 matrix, and from k to j, I have an entry in the A matrix. So this gives me A^3, so you can do the same calculation as before and you can come up with a new matrix. So I started with A, I did one pass over A and I got A^2. Now using A and A^2, I can get A^3.

(Refer Slide Time: 13:07)

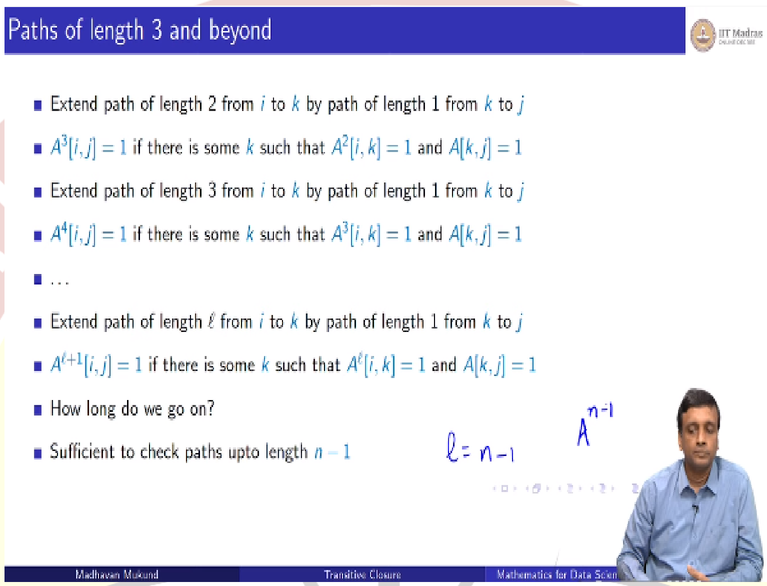


So, now I can go from 3 to 4 in the same way, if I have a path of length 4, then it can be decomposed as a path of length 3 followed by 1 edge. So, I can take the entries in A^3 and combine them with entries in A, so I can look for a k such that A^3 i, k is 1 and A k, j is 1. So, this now gives me A^4 i, j, which will be 1 provided there is a path of length 3 from some i to some intermediate k, followed by a path of length 1.

Now, you could also do this as 1+ 3, you can do it as 2+ 2, but let us just follow this general rule, where we break it up into one less or + one. So, in general, if we keep going, right, so if I want to, I already know paths of length l, and I want to extend it to l+ 1, then I will say that A_l+ 1 i, j is 1. If I already know that there is a k for which there is a path of length l and I can extend it by 1 edge, a path of length 1 from k to j. So, I can go from i following some l steps to k and then I can go from k to k in one step. So, then I can go therefore from here to here in l+ 1 steps.

So, we just do the same thing again and again. The first time we are doing A combined with A, second time we are doing A^2 combined with A, in general we do Al combined with A and each time I will go from here I get A^2, from here I get A^3, from here I get A^l+ 1 and so on. So, I can keep on building this matrix, which captures longer and longer paths.

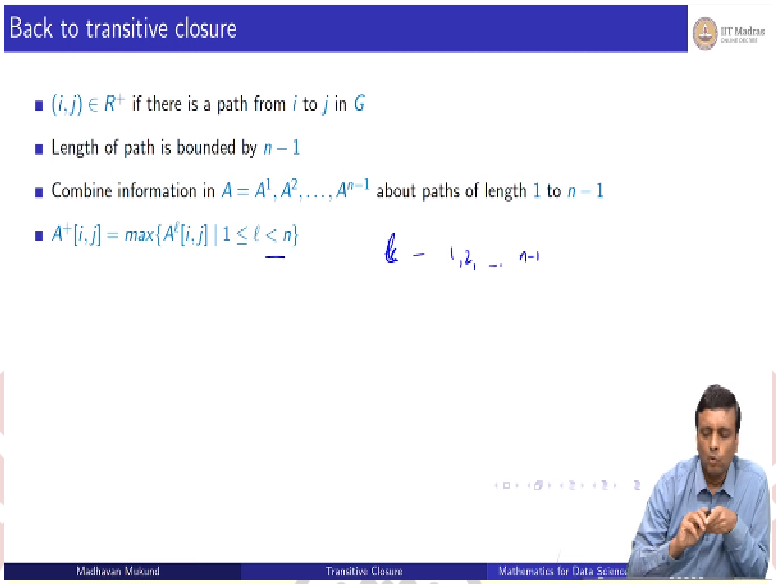
(Refer Slide Time: 14:44)



So, now where do we stop? How long do we go on? Well, here we know that if there is a path at all, then that path cannot have more than n-1 edges, because once I have traversed n-1 edges, I have seen n-1 different vertices other than the starting point and therefore, anything beyond that must repeat a vertex, so there must have been a shorter path.

So, therefore, if there is a path at all from i to j, it cannot have length more than $n-1$, so I can stop with. So, once I have computed A to the $n-1$, right, I have got everything of interest, I have got all paths of length 1, 2, 3, 4, up to n-1. And any path which is longer than n-1 cannot be new. I mean, since it cannot contribute any new information to me about whether or not i and j are connected by the relation or not.

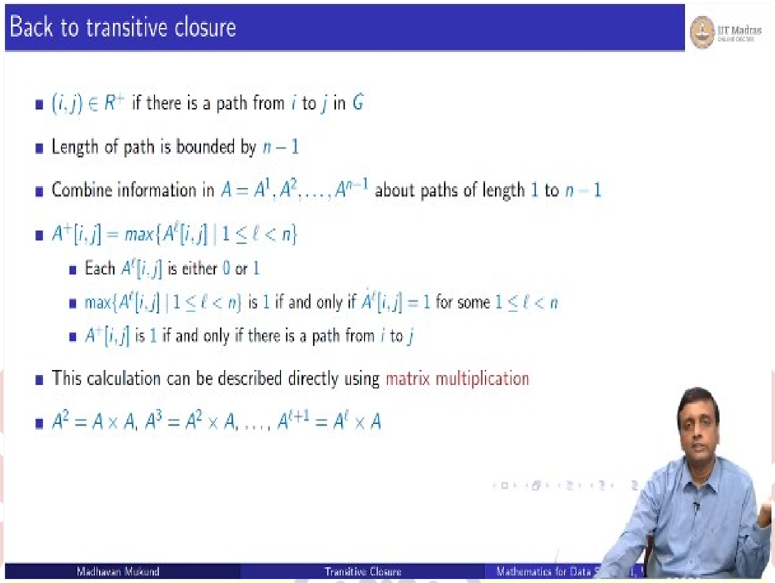
(Refer Slide Time: 15:37)



So, remember that the reason we are doing this is for the transitive closure. So, we said that i, j is in the transitive closure R^+, if there is a path from i to j in the corresponding graph for R. And we have observed many times that the length of this path is at most $n-1$. So, therefore, we can combine all this information, right, so, we have the original A which is the same as A^1, right, path of length 1, then from that we computed as A^2, then A^3, A^4 and so on and up to the A^{n-1}.

So, I have this n-1 matrices, which gives me all information about paths from length 1 to length n-1. So, what do I want to do? I want to say that there is an edge from i to j in the R^+ relation. If there is an edge somewhere in one of these, right, and I am going to write it in this complicated way, I am going to say it is a maximum of the i, j entry in all the matrices from k =1 to k =n-1, notice that this is strictly less than it. So, k goes from 1, 2 up to $n-1$ because it starts at 1 l, A to the l sorry, l goes from one to n-1, right.

(Refer Slide Time: 16:50)



So, this is, so, what does this mean? So, notice that each and each entry is 0 or 1. So, every entry of these, these are all 0, 1 matrices, either there is a path of that length or there is not a path of that length. So now, when I am taking this max, it is basically checking if all of them are 0, that is there is no path of length 1, there is no path of length 2, there is no path of length 3 and so on, the max is also going to be 0.

So, there will be an A+ entry which is 0, there is no path, but if there is a 1 anywhere, right, in any one of those l positions from 1 to n-1, if any one of them is 1, then the max will become 1, if there are many paths, there are path of length 3 and 7, it will still be 1 because max of 1 and 1 will remain 1.

So, by taking max we are just recording is there at least 1, 1 in that sequence or not? Sequence meaning across all these matrices in the ij th position is there at least one of these matrices which has position value 1 at i, j. If so, the max will give me 1, if all of them are 0 max will give me 0.

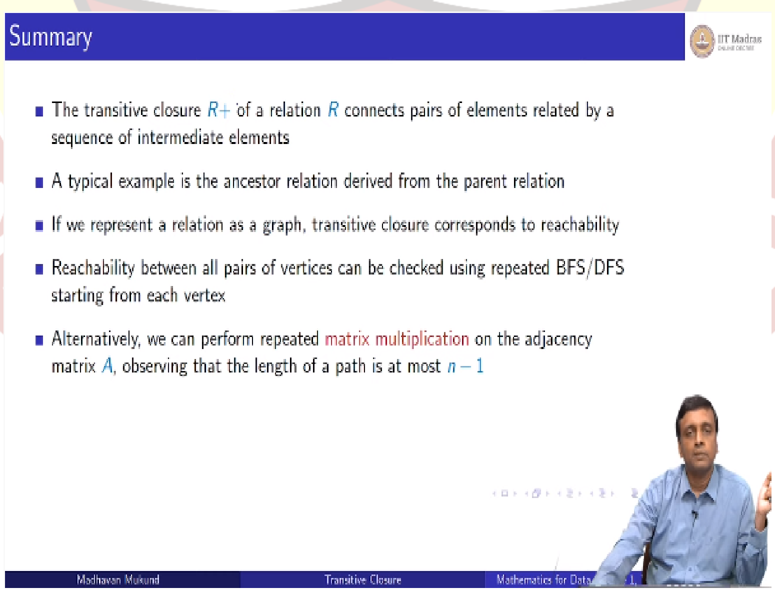
So, in that sense, this A+ entry captures the fact that there exists some length path between 1 and n-1, between i and j and if it is 0, it means there is no such path, right, and we know that if there is no path of length n-1, there cannot be a longer path because if there is a path it must have at most length n-1, anything longer than that will be looping and will be redundant.

So, therefore, A+ i,j is 1, if and only if there is a path from i to j and in particular, this path must always be bounded by length n-1. So, what we have done, we can actually reformulate it in a way that is called matrix multiplication, which we will not do right now.

But it is important to know that this, what we did is a very tedious calculation rows and columns and all that is actually a very standard operation on matrices. So in this form, we can write it as a sort of multiplication of matrices. It is not exactly what I have written here, but for the purpose of this lecture, this is fine.

So, you can actually believe that this operation, the reason we are doing this with matrices is that this operation on matrices is actually a standard mathematical operations on matrices. So, though we have done this column and row chasing explicitly, saying we look for a k here, we look for a k there, actually, you can actually do it directly as a matrix operation. So therefore, it is a very standard operation is not something new.

(Refer Slide Time: 19:09)



So, to summarize, the transitive closure tells us so, this would be an R^+ on top, the transitive closure tells us whether there is a sequence of intermediate elements which connect two elements, right. So, I have to start at p and go through multiple R edges to reach q, an example of this was our ancestor relation. So, the ancestor from the parent, so a sequence of parent edges generates the ancestor edge.

So clearly, since we visualize relations as graphs, this corresponds to a path, right, and in general, these are directed edges because these relations are not assumed to be symmetric, like the parent relation is certainly not symmetric, if A is parent of B clearly B is not a parent of A, right. So therefore, in general, you follow a path in a directed sense, and this is just a reachability question in graphs.

And we know that we can do this by repeatedly doing BFS and DFS from every starting point, but what we have seen in this lecture is that alternatively, we can take the adjacency matrix and do a form of matrix multiplication, we can do a form of matrix multiplication to go from A to A^2 to A^3 and so on and stop with A to the n-1 because A to the n-1 records paths of length n-1 which is the maximum length path, which is useful to us to find out whether two edges, two nodes are connected and once we have got this we can take a look for a 1 in any one of these n-1 matrices and if so, declare A+ ij to be 1.