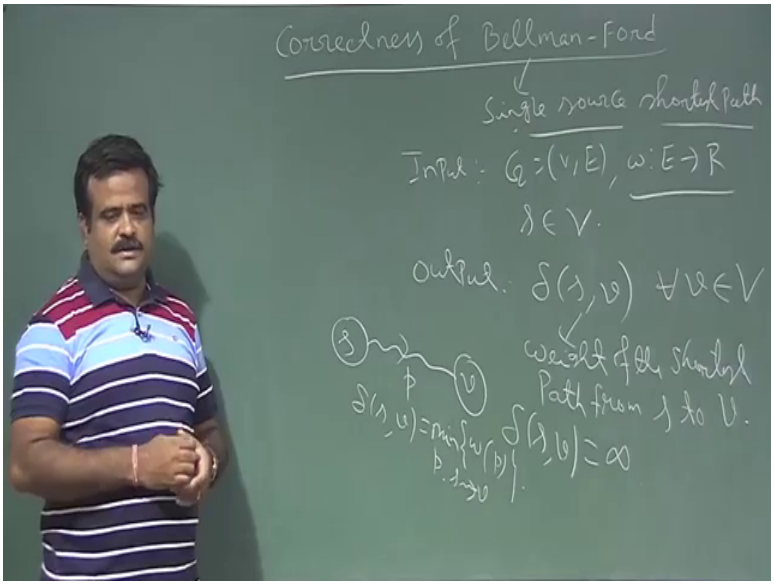
An Introduction to Algorithms
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 46
Correctness of Bellman Ford

So we are talking about bellman ford algorithm. So, today we will discuss the correctness of the algorithm. So, what is the bellman ford algorithm it is to find the single source shortest path single source shortest path.

(Refer Slide Time: 00:27)

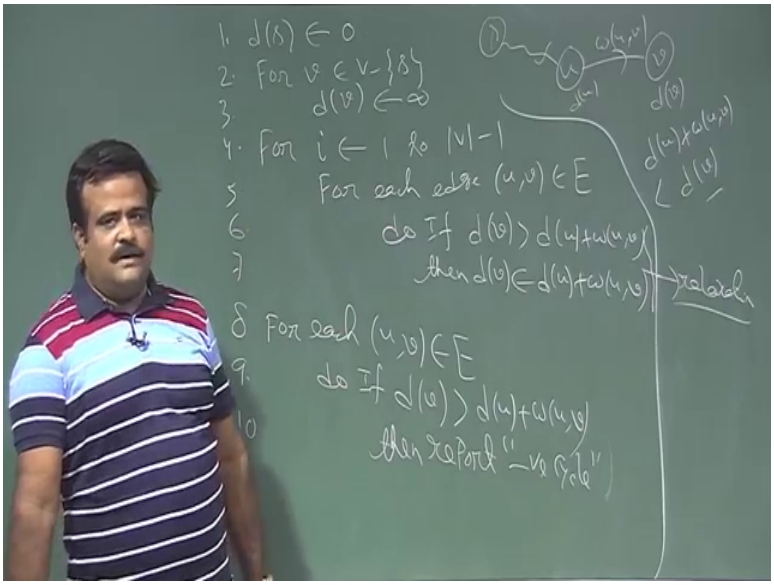


So, we discuss this algorithm in the last class, but just to recap. So, the input is a graph directed graph. And a weight function and here we have no restriction of the positive weight. So, we are allowing to have the negative weight. So, there is a chance of negative cycle, but our algorithm can handle that. And we have a source vertex which is which is also another input. And the output will be the delta of s comma v. So, this is the weight of the shortest path, weight of the shortest path from s to form source to the vertex v. So that means, you have a vertex s and you take any other vertex v. So, there are many paths from there could be many path from this. If there is no shortest path then we denote delta of s comma v as infinity. So, this can happened either there is no physical path from s to v, or there is a path from s to v, but there is a negative cycle from in that path. In that case the shortest path will not exist.

So, this is one of the one of the condition where the shortest path will not exist. So, this is for there is no path one case, and the another case all though there is a path between s to v, but there is a negative cycle in the in the path. So, then there will be a no shortest path in that case we denote this by this. So, this is the path p which is the shortest path. So, delta of s comma v is basically weight of the. So, minimum weight among so, all the path which is from s to v. So, this is the shortest path weight form from s to v. So, you have to find shortest path from s to any other vertices. So, that is the problem ok.

So, let us just recap the bellman ford algorithm then will talk about correctness of this. So, for bellman ford algorithm we are just having initialization step. So, this is the pseudo code for this algorithm we are assigning degree of s to be 0.

(Refer Slide Time: 03:19)



S is the source vertex which is the input and for any other vertex. We are assigning the degree to be infinity because we have an explored it. And then so, and then for I is equal to 1 to v minus 1 times this is outer loop and we are doing the branch of relaxation. And for each edge and that is it is loop for each edge u v belongs to E. What we are doing we are? So, we take a edge u v, u to v direct edge and we are checking whether we could relax this vertex or not. So that means, this is having a d v which is the shortest distain estimate s to that vertex.

So now we have now for this edge we have new path we have s somewhere here. So, s to this node this is d u then w u v. So, we have a path now we have a path s to v is like this,

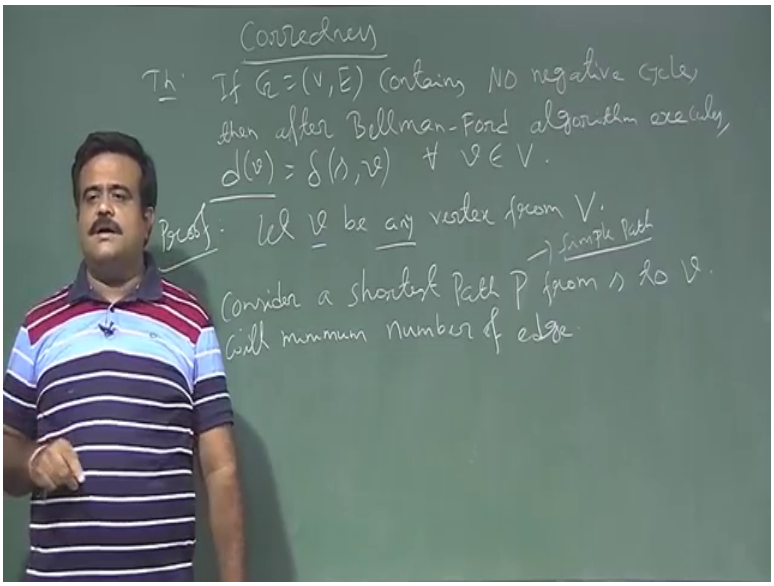
d u plus u v. So, if this is less than the path we have then we must relax this. So, this we have to do if d v is greater than d u plus w u v if d v is greater than d u plus w u v then we must relax that a vertex v then d v must be relax. So, this is the relaxation, this step is the relaxation step. This is the relaxation step. So, this is 6 7 and then finally so, this is this should give us if there is no negative cycle then there will be shortest path.

So, this should give us the shortest path if there is no negative cycle any way we will proof the correctness. And after that if there is a negative cycle then it will keep on relaxing some of the vertices. So, for that we need to have a last checking. So, this is the for each u v belongs to E we just do this final checking do if d v is if v could be further relax then there is a there is a problem; that means, the negative cycle exist in the path. Then we put negative cycle that is it. So, this is the this is the pseudo code for bellman ford algorithm.

So, for this purpose in this in the last class you have taken the one example. So, there we have to execute this for v minus 1 times this outer loop. So, why v minus 1 time will we will discuss this in the correctness. And also we have to do it inner loop is for all the all the edges. So, what we did we did we have edge numbering. So, we have following the same numbering for all the all the outer loop. So, we do the edge numbering then we following this numbering for and then we are executing this relaxation step and finally, when we exhaust if it is converging; that means, there is no negative cycle and it must converge by this many times I mean v minus n times.

So, let us just talk about the correctness of this algorithm. And why it is? Why the outer loop is v minus 1 time? So, let us talk about correctness of bellman ford.

(Refer Slide Time: 07:45)



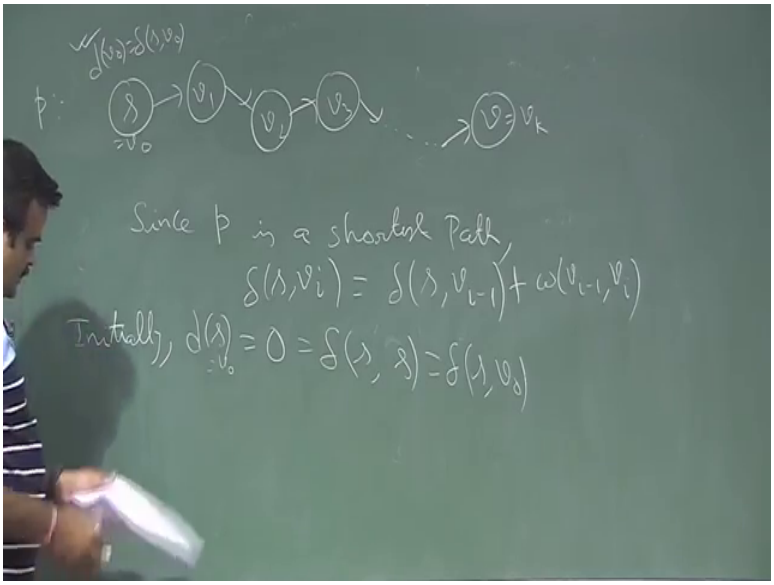
So, what is this telling? This is in theorem if G of v E is graph directed graph which content no negative cycle, no negative cycle then, then after bellman ford after bellman ford executes algorithm executes. We have the degree basically will give us the deltas, for all v belongs to v. So, this is the correctness of the bellman ford algorithm. What it is telling it is telling us if there is no negative cycle in the path in the graph So that means, the shortest path will exist now if the shortest path will exist then, then the after execution of the bellman ford for that many time for v minus 1 time by order of v minus 1 time, it should give us the shortest path weight. And that is basically coming from the degree of each of the vertices. So, that you have to prove and why v minus 1 time that will see soon ok.

So, how to proof this? So, to proof this we are taking let v be a any vertices vertex form capital V. And we will see delta d u v of v is basically delta of x comma v. Now consider a because there is no negative cycle. So, we consider a shortest path, shortest path which is denoted by p form s to v with minimum number of edges minimum number of edges. This is important, because we may have cycle. So, this path is simple path consider a shortest path. So, this p is a simple path what do you mean by simple path. So, we may have cycle in the path, but since there is no negative cycle. So, that cycle has to be positive or 0. So, you can just ignore that cycle so that means, in the path p there is no repetition of the vertices. So, that is possible because we are we are assuming there is no negative cycle. So, if at all there is a cycle in the path that cycle could be positive or 0.

So, we are talking about shortest path. So, we may ignore that cycle will not visit that cycle will not hop in that cycle. Because that cycle has will not get a any benefit to us because that cycle is positive cycle or 0 cycle. So, so we will just ignore that. So, p the path p will be just the vertexes are dishing. So, there is no repetition of the vertices ok.

So, let us consider this path p. So, let us write the path p.

(Refer Slide Time: 11:57)



So, we have s which we have denoting by say v 0 and we have v over here. So, this is our path p. So, it may be visiting some vertex v 1 v 2 v 3 and so on. Dot, dot, dot and this is say v k. So, there are say k elements dot, dot, dot, dot, dot v k elements. K number of vertex in the path. And since p is a shortest path, since p is a shortest path. So, what we have we have? Delta of s comma v i is basically delta of s comma vi minus 1 plus w of v i minus 1 then v i ok.

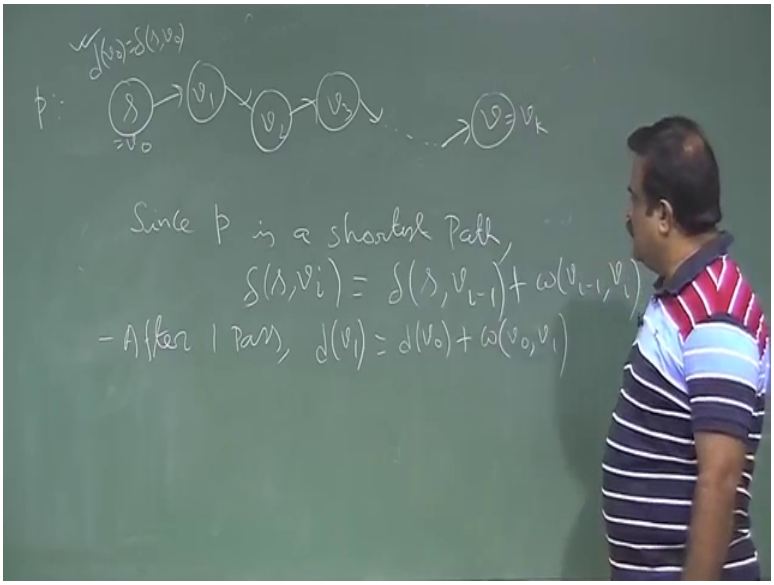
So, if you take a vertex up to v i. So, the this is optimal substructure problem. So, up to what optimal in the last class we have seen what optimal substructure problem. If your path is if your path is a shortest path of a if p is shortest path of a graph then any sub path we will give the shortest path of that vertices up to that. So, that is the optimal sub structure problem. So, if you take a sub path up to v i. So, that will be the shortest path of that. So, that is basically v i minus i that is basically, because this is the this is a path this is a shortest path from s to v i minus 1 and this is a shortest path form v i minus 1 to v i.

So that means, if we sum it up that should give us the shortest path form, because that is the sub path of this path. So, that should give us the shortest path from s to v i ok.

So now initially. So, we are putting initially degree of s is 0. Now that is basically delta of this s is v 0 delta of s comma s. So, s is nothing by v 0 delta of s comma v 0. Because why this is true? Because there is no negative cycle. So, what is the shortest path weight form s to s if there is no negative cycle? Then the shortest path from s to s the best thing we can do we remain at s. So, that is basically d of degree of s initialization this is the initial step. So, this is the initially, initially, initially we have this. So, initially this is done. So, degree of v 0 is basically delta of s comma v 0. So, this is done in the step 0 or initial step ok.

Now So, after 1 pass. So, we are having if we recall the algorithm we are having a loop for in the bellman ford algorithm. We are having a loop for i 1 to v minus 1 and for under this loop we are checking all the all the edges and we are trying to relax vertex b. So, this loop will start from I is equal to 1 i is equal to 2 like this. So, this is I is equal to 1 means first pass like this one pass i is equal to 2 means second pass like this. If you consider like this. So, after first pass. So, after first pass. So, what will happen. So, initially this is done. So, after first pass.

(Refer Slide Time: 16:11)



So, after 1 pass. So, what will happen? This is a edge in the graph we do not know the numbering of this h, but this is a edge. So, this edge number will come in some order

because in a in a inner loop of a bellman ford we are doing this for all the edges. So, this edge will come this edge number will come in some order may not be in the first order first edge number 1. So, one this edge will come in that case what will happened? The degree will be d of v will be basically will change because everything is initialized by infinity.

So, degree of v 1 will change to d of v 0 plus w of v 0 v 1. So, this is after pass and this is nothing but what this is nothing but delta of s comma v 0 plus w of v 0 v 1 and from this result this is nothing but delta of s comma v 1. So, after first pass this is done v 1 is basically delta of s comma v 1 this is done after first pass. Because this edge number will come in some order it may not be in the first sort. So, this edge number if it is 1, then we are luck we will we just change this by in the first sort of way, but we do not know how this ordering will come. So, this is fixed this is this is done this is done after first pass.

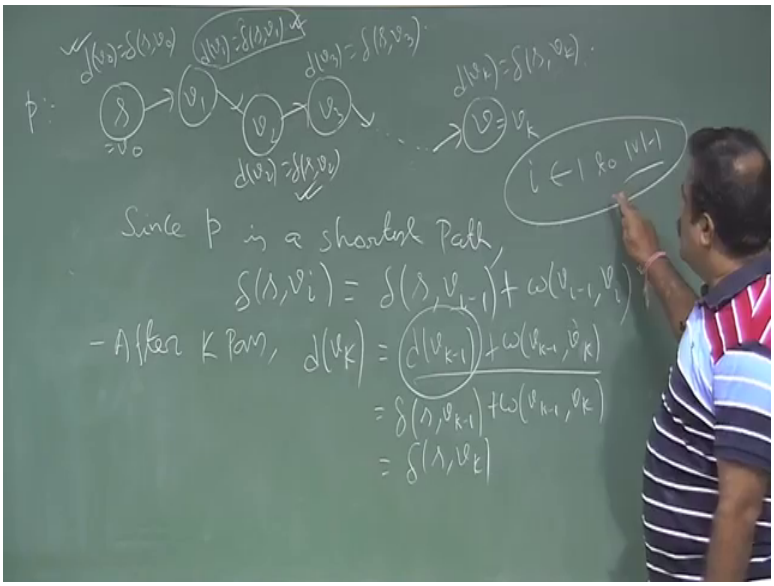
Now, similarly after second pass what will happen? So, after second pass. So, once this is done in the first pass then the second pass this is one edge in the second pass means I is equal to 2. I is equal to 2 we are doing for all the edges again. So, this is one of the edge. So, this edge number will come in some order. So, once this edge number will come, then it will do what it will change this. So, the degree of v 2 will change to d of v 1 plus w of v 1 comma v 2. So, this has to be change. So, this is basically now this is already done. So, this is basically delta of s comma v 1 plus w of v 1 comma v 2. And So, this is from this theorem this is nothing but delta of s comma v 2. So, this is done after. So, this is done after second pass. So, this is basically d of v 2 is basically delta of s comma v 2. So, this is done in the second pass ok.

Now, similarly so suppose on number is like that edge numbering this is number 1 edge this is number 2 edge like this. Then in the first part first pass itself it is done, why because first pass if this is edge number 1. So, we will start with this edge or if this number is before this. Then this is done in the first pass. And then when this edge number will come after this then this will be done, but we cannot assume na because this s is v is any vertices. So, this ordering we do not know. It depends how we are numbering them so, but any way after second pass this number will come and when this number will come this is already fixed. Now was this is fixed when this number will come this will make this fix ok.

So, this is done after second pass. So, once this is done after second pass. So, in the third pass what will happened? So, after third pass. So, after third pass So, this degree will change degree of this will be. So, this edge will be this vertex will be relaxed degree of v 2 plus w of v 3. So now, this is already done. So, so this is basically delta of s comma v 2 plus w of v 2 comma v 3. Now again by this theorem by this result we can say this is nothing but delta of s comma v 3. So, this is done after third pass. So, after third pass after I is equal to 3 this is done.

Even this could be done in the first part if in the first pass I is equal to 1 if the ordering is like, this if this is before this order before if this is after this order, then it could be done in the in the I s equal to 1 itself. But any way you do not know the ordering I mean how we numbering, So that is why we have to do like this. So, d 3 will be delta of s comma v 3. Now So, this So, we will continue and after the k th pass. So, what will happened? So, after the k th pass. So, this edge will be if it is not yet relaxed, if it is not yet relaxed and reach to the this final mode.

(Refer Slide Time: 22:18)



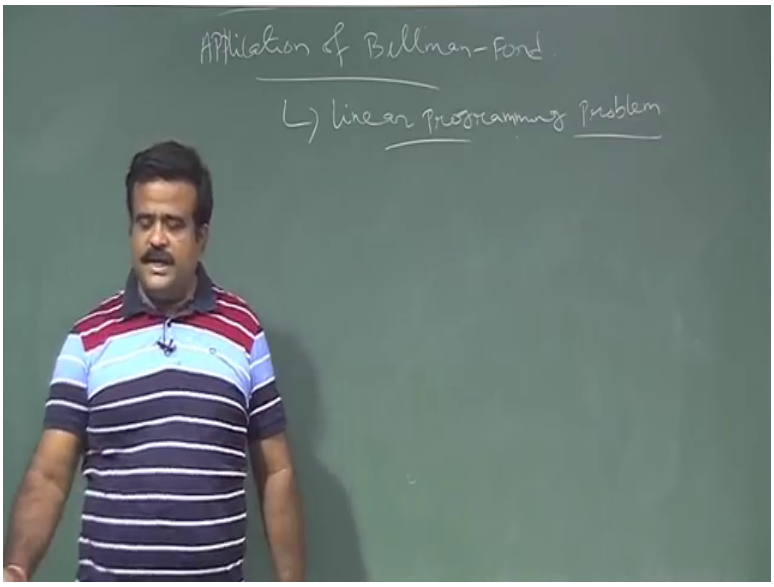
So, we have to relax again I mean it will be get relax in the k th pa pass at least.

So, k th pass if this d of v k minus 1 is plus w of v k minus 1 comma v k. If this is if this is of this is less than that then you must the already the degree was having. So, we must relax this and this because this is already done in the k minus 1 path. So, this will be basically delta because this is delta of s comma v k minus 1 plus w of v k minus 1

comma v k. And this is nothing but d of v k minus 1 plus. So now, this is nothing but delta of s comma v k. So, by this result. So, this is done after k th pass. So now So, this is after k th pass those this v k will be basically d of v q is basically delta of s comma v k. So now, this is the correctness.

Now, that is why it is why the outer loop of bellman ford is one to order of v minus 1. Because this is because this. So, v we do not know the why it is v. So now, how many vertices are there this is a simple path? Because we there is no negative cycle. So, every path is single path. So, this vertices distinct. So, how many vertices we can have in the middle at most? So, there are at most v vertices. So, there could be v minus 1 vertices in the middle. So, every path from s to any other vertices can content at most v vertices in the middle. So, that is why that the outer loop for the bellman ford algorithm for up to v minus 1 because this path any path any part of such p content the number of vertices is v minus 1. So, that is the reason we run that outer loop up to v minus 1. So, this the correctness of bellman ford algorithm. Now in we will discuss some application of bellman ford algorithm, like how to solve the linear programming problem. So, next topic will be some application of bellman ford algorithm.

(Refer Slide Time: 25:04)



So, well we will talk about the how to solve the linear programming problem. Not the general problem general linear programming problem, but a particular version of a liner

programming problem which is called liner constraint, but that will discuss in the next class.

Thank you.