

Data Structures and Algorithms
Dr. Naveen Garg
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture – 35
Correctness of Dijkstra's Algorithm

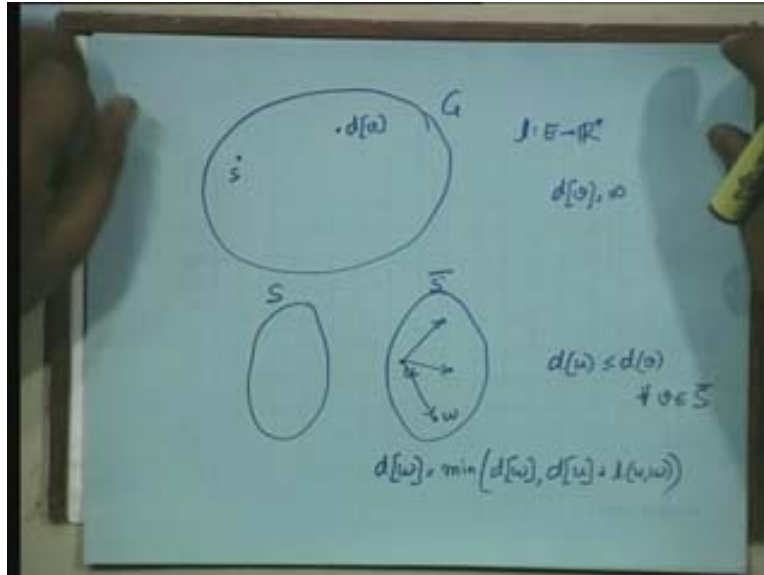
In the last class we looked at dijkstra's algorithm for computing the single source shortest path. The algorithm I do not recall if I did tell you the name of the algorithm. It is called dijkstra's algorithm. What remains to be done? We have seen what the algorithm is. Hopefully you all understood, I am going to be revising it any case today. We had also argued how to implement the algorithm and what its running times would be. The implementation was very similar to prim's algorithm computing minimum spanning tree with some small modifications. But we have not argued why the algorithm is correct? Why the dijkstra's algorithm is correct?

As a quick recap of what is being done in dijkstra's algorithm. We are given a source vertex, we are given a graph. You are given a length function on the edges and the edges are non negative. You can have zero edge lengths but you cannot have a negative edge length. That is important and we will see why. What is happening at each point? With each vertex we have a distance label associated with that vertex $d[v]$. Today we have to first understand the semantics. What do these numbers represent? Initially $d[v]$ recall is infinite for every vertex. $d[v]$ is an upper bound on the distance of v from s . And this quantity with every step only decreases and it never increases.

At a particular step what does the algorithm do? At any point we have two sets, the capital s and s complement. The vertices which are in s , for them we have computed there actual shortest path distances from s . We have already computed that. And the vertices which are in s complement, for them we do not know the correct distance. We only have an upper bound on the correct distance. At a particular iteration what do we do? We take the vertex for which the $d[v]$ is minimum.

Suppose this is the vertex u , so $d[u]$ is less than or equal to $d[v]$ for all v in s complement. (Hindi Conversation) signifying the fact (Hindi Conversation). And then we are going to move u across. You will go from s complement to s . We will also have to make this claim to argue correctness. That is at this point (Hindi Conversation) that is the correct shortest path distance of u from s . When we move across how do we update the other labels? If there are 3 edges going out of u to other vertices in s complement (Hindi Conversation) What is the new distance label of w ? $\min d[w], d[u] + \text{length of the edge } [u, w]$.

(Refer Slide Time: 06:38)



This is how we update the distances (Hindi Conversation). As you can see this operation shows that $d[w]$ value will only decrease, it cannot increase. Because it is the minimum of two quantities one of which is $d[w]$. $d[w]$ value will only decrease. So if these distance label were sitting in a heap then all you have to do is to do a decrease priority operation on the heap. When you are removing the minimum that correspond to the delete min operation. This is just a recap of what we have done in the last class. We have seen all of this before. (Hindi Conversation) that is the algorithm we keep doing this, till every things moves from here to here and then we are done. What is the initial composition of s and s complement? Initially the only vertex in capital s is little s and every thing else is in s complement.

Let us first understand given a vertex u , what does $d[u]$ means. Given a vertex u in s complement, this is s and little s is here (Refer Slide Time: 06:50). What is the meaning of $d[u]$? What connotation or what semantic can I associate with $d[u]$? **Shortest length of a path including vertices only from... yes.** So claim is $d[u]$ is the shortest path from s . Let us look at all path from s to u which use only vertices of s and at one point that take one edge across and reach u . (Hindi Conversation)

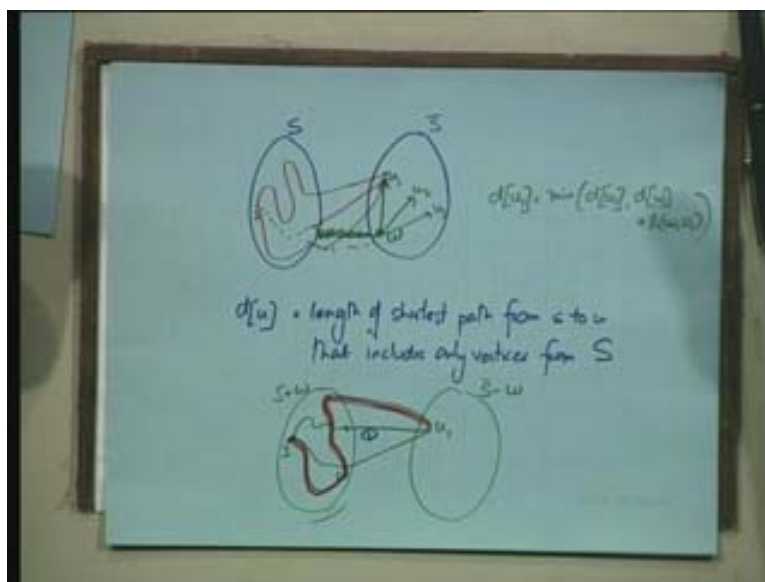
There can be many such paths. This is one path may be there is a second path which goes like this. There may be a third path which goes like that and so on. There could be again a whole lot of such paths. Let us compute the length of all those paths and let us take smallest amongst those. That is $d[u]$, so formally I could try $d[u] = \text{length of shortest path from } s \text{ to } u \text{ that includes only vertices from the set } s$. (Hindi Conversation) except for u which is the last vertex, the other intermediate vertices will only be vertices from s . That is the meaning of $d[u]$, that I clime is the meaning of $d[u]$ and let see whether this is really true.

How will we prove that this is true? The best way to do this is to prove it by induction. Make sense to try and prove it by induction. We will assume that at this point this is true. The numbers or that d on these vertices do reflect this quantity. And then we will say when I move one vertex across then the new d 's on the vertices continue to reflect the same thing. And then we will show that the base case is also true and that will establish this claim. If it is true now I move a certain vertex across. Let say vertex w which is here, moves across. As a consequence I might end up updating the distance label of u . (Refer Slide Time: 10:14) so this edge is not there, this is just signifying that this vertex moves across. Let say u_1, u_2, u_3 are the 3 neighbors of w that is out neighbors.

So we are going to update u_1 to $d[u_1] = \min [d[u_1], d[w] + l[w, u_1]]$. This is how we are going to update. Why is this correct? Why is it that after this the new value of $d[u_1]$? May it is the new value, may be it is the old value, still satisfies this thing. If $d[u_1]$ remains unchanged then that means the earlier value was the smaller one. So what is that happening? The w moved across, let me draw the picture s complement minus w and $s+w$, w is here, and u_1 is here. Not another path, but I should say let me look at all the paths from s to u_1 . Now those paths, what could they be doing? So I have to look at all path form s to u_1 which only includes vertices from this side. (Hindi Conversation) Either the paths do not include vertex w at all and they go directly like this but then the length of such a path is already captured in the earlier value of $d[u_1]$. The $d[u_1]$ is smaller than the length of all such paths or such a path includes w . One possibility is that it includes w , that it visits w and then goes to u_1 .

Third possibility, it goes to w goes to some other vertex and then goes to u_1 . Let us look at these 3 things. We are saying that either it does not visit w at all which is the first case. Then what remains is, it visits w . It touches w . What touches w ? We are looking at the shortest path from s to u_1 which includes only these vertices.

(Refer Slide Time: 13:51)

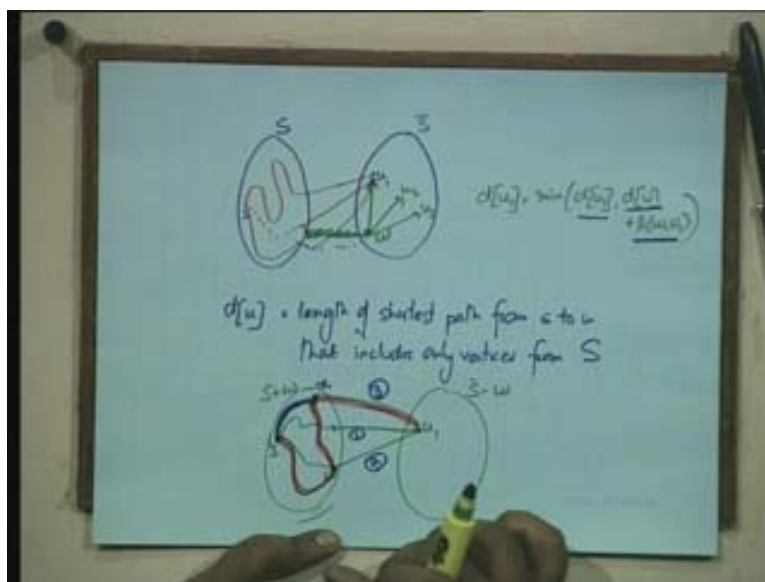


So the other case is that it touches w . And the moment it hits w after that it goes directly into u_1 . This is the second case. Third is it comes to w then it goes of some where else that it goes to u_1 . You are claiming that the third case cannot happen. What do you mean by cannot happen? The claim is that this path has length at least as large as $d[u_1]$.

We have to look at these three cases one after the other. The case one is covered. Case two is also covered. Why because this path comes up to here (Refer Slide Time: 15:06) and then I take this edge. If this is the shortest path from s to u_1 and it goes via w like this then this part has to be the shortest path from s to w (Refer Slide Time: 15:26). That is equal to of length $d[w]$ by induction hypothesis. And then this plus this length would be the length of this path from s to u_1 . That is getting captured here. We are accounting for the length of such a path. What remains is this case three. Why cannot we have this? Let say this vertex is x . You are wrongly concentrating on w , it has to be x . We have to look at x . The x was already there. We already knew the shortest path from s to x and that was equal to $d[x]$, the label on x . It does not involve w , we already have that information.

If this is the shortest path from s to u_1 then s to x is also the shortest path from s to u_1 . What is the blue line that I have drawn? This is this the shortest path from s to x . The x was already a path of s which means we already had the shortest path from s to x . And the shortest path from s to x is using only vertices of capital s , not using w at all and that path sits here. If this path which goes from s to w then from w to x and then from x to u_1 is the shortest path from s to u_1 using vertices only of $s + w$. Then this path from s to x followed by x to u_1 is also the shortest path. If it is also the shortest path then we have already accounted for it in $d[u_1]$. $d[u_1]$ is only less than this quantity, because what is $d[u_1]$? $d[u_1]$ till now is the length of the shortest path from s to u_1 using vertices only of capital s . So this path that is now I have created does not use vertex w at all. So it becomes something like case one, except that you have to make this argument.

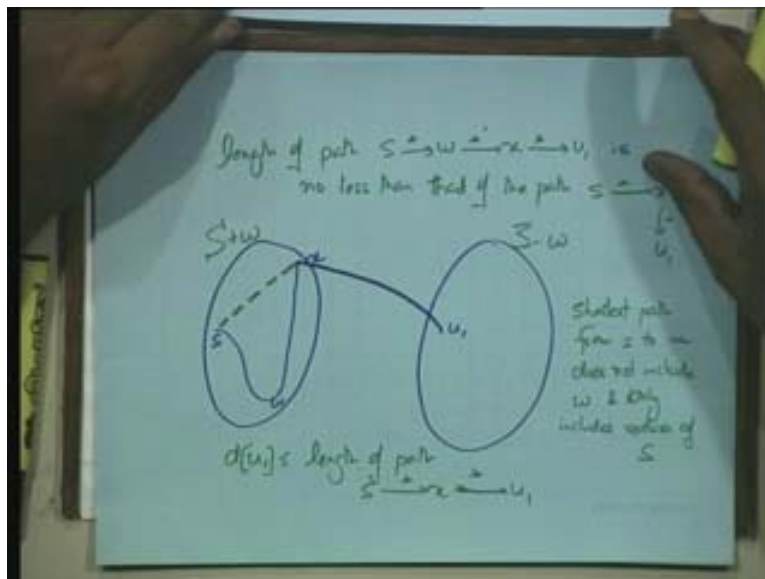
(Refer Slide Time: 18:52)



So may be all of this is not very clear. So let us try and do it more completely. We are looking at s, u_1, w is the vertex which has just come in. And we are looking at third case, when shortest path from s to u_1 goes via w to a vertex x and then it jumps across to u_1 . Now the length of this path, which is s to w plus w to x plus x to u_1 . The claim is that the length of this path is at least as large as the length of this path from s to x , followed by x to u_1 . Let me write it down, length of path s to w to x to u_1 is no less than that of the path s to x to u_1 .

What is the path from s to x , I am taking? The shortest path from s to x , that does not include w . Shortest path from s to x does not include w . Why, because w just came in and x was already there. By our induction hypothesis we had already found out the shortest path from s to x . That is the path which is lying completely in S . Let me write that part also. The shortest path from s to x does not include w and only includes vertices of capital S . This is capital $S + w$ now, this is capital S complement minus w . Capital s was the part before we moved this thing. This path which is going like this (Refer Slide Time: 21:28), has length at least as large as the length of this. But what do I know about this? About this I know that $d[u_1]$ is less than or equal to this u_1 , length of path s to x to u_1 . Already I know that, because $d[u_1]$ was the smallest possible path which used only vertices of s . This is only less than that and since this is less than this, $d[u_1]$ is less than or equal to this longer path. So I can effectively ignore this thing, I do not have to consider it, because it is already included in my $d[u_1]$. $d[u_1]$ is already a smaller quantity than that. So all such paths are captured in $d[u_1]$. $d[u_1]$ is the quantity only smaller than such paths. The one path which is not captured is such a path which goes to w and then takes an edge out that is case 2. For that we have included here. And this smaller of these 2, will give me the smallest path from s to u_1 which only includes vertices of $s + w$ (Hindi Conversation).

(Refer Slide Time: 23:24)

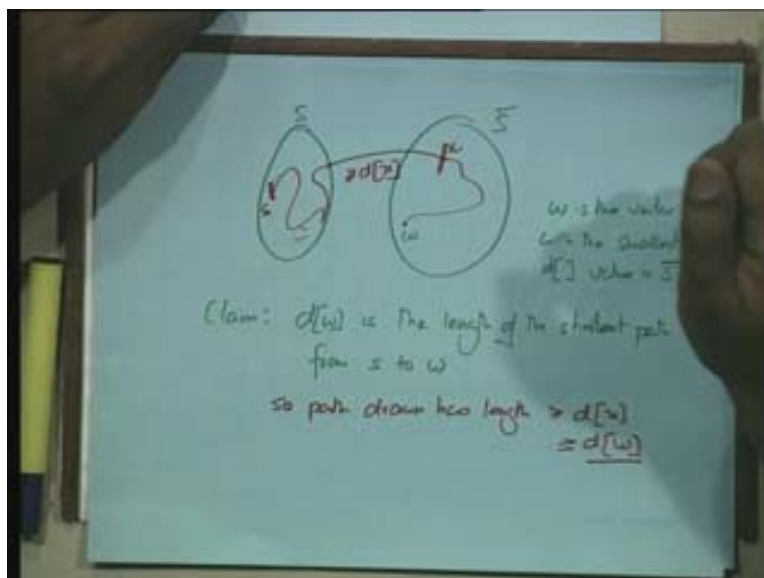


That is a semantic and once you have the semantic in mind then it is easy to prove, most of the other things. What is the meaning of those distance label? Once you have that in mind every thing else are very simple to follow. Are every one convinced that d distance label reflect this kind of length. Why are we justified in moving the smallest label vertex from here? That is what we are doing at each step. At each step I am taking this vertex which has the smallest label and moving it across and claming that it has the right number on it. Why am I justified in doing this? What we are claming is $d[w]$ is the length of the shortest path. What is w ? w is the vertex which has the smallest label.

The w is the vertex with the smallest d value in s complement. The claim is $d[w]$ is the length of the shortest path from s to w . Why is this claim true? Proof by contradiction, suppose this is not true. If this is not true what does that mean? That there is a shorter path form s to w . Shorter than what? Shorter than $d[w]$. What will that path do? That path will visit some vertices here (Refer Slide Time: 25:47) and at some point it has to jump across. And then it has to come to w , may be it jumps back and then comes again. It can do a whole lot of crazy things, but it has to at least jump across once.

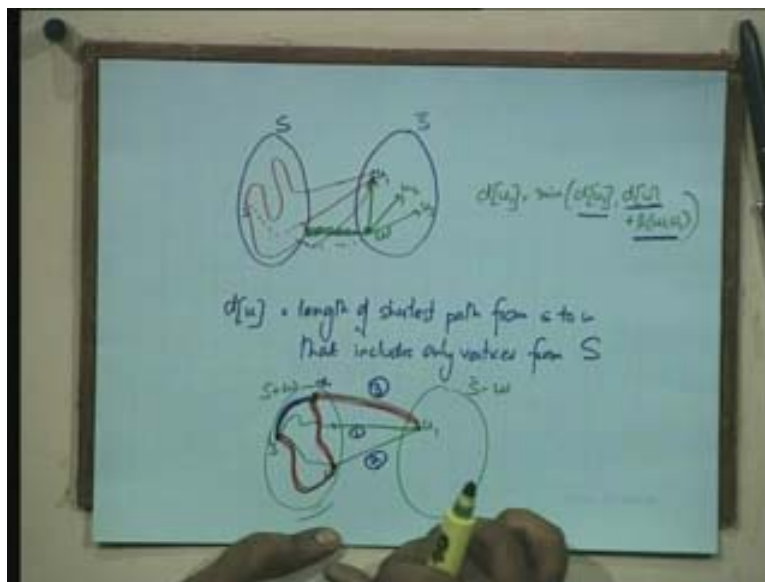
If it jumps across once, let say it jumps across and reaches vertex x . What is the length of this portion of the path? (Refer Slide Time: 26:16) greater than or equal to $d[x]$. That is our semantic that we have associated with $d[x]$. So this part of the path is at least $d[x]$. (Hindi Conversation) Why because we said, the shortest path from s to x which uses only vertices of s , has length $d[x]$. This is the path which uses vertices only of s . So it has length at least $d[x]$ or more. That is why I put greater than or equal to. This part of the path whose length is at least $d[x]$ which means the entire path has length at least $d[x]$. Here I am using the fact that edge lengths are non negative. So this entire path that we have drawn has length at least $d[x]$. So path drawn has length greater than or equal to $d[x]$, which is greater than $d[w]$. It is greater than or equal to, may be they were at the same distance.

(Refer Slide Time: 26:43)



So this path has length at least as large as $d[w]$ (Refer Slide Time: 27:50). Which means it is not a shorter path. We started off by saying that there is a shorter path than $d[w]$. There is a path whose length is strictly less than $d[w]$, so that is not the case. If there was such a path it should have a length, greater than or equal to $d[x]$ it cannot be strictly less than $d[u]$. You said this is the best path. All I am saying is this part of the path has to have length at least $d[x]$. If it were the shortest path then it would have exactly $d[x]$. But maybe, you did something I do not know, it is at least $d[x]$. (Hindi Conversation) there is no path which is smaller than length $d[w]$. There is no possible path at all. There is no path from s to w of length strictly less than $d[w]$, which means $d[w]$ is the length of the shortest path from s to w .

(Refer Slide Time: 30:37)



And so we are justified in moving this vertex across because we have found the length of the shortest path from s to w . What is that path? That path would include some vertices from here and then jumps across. We are justified in moving this vertex. Let us recap the argument. This claim is okay (Hindi Conversation) we were proving this (Refer Slide Time: 30:07) length of shortest path from s to u that includes only vertices from S equals $d[u]$. This we were proving using induction and when proving this induction statement we required that (Hindi Conversation) we have found a shortest path from s to x already. And that we have proved now which is shown in the slide that is (Refer Slide Time: 26:43) (Hindi Conversation) Induction is being applied on the number of elements of S . So what is the induction statement? What should be the induction statement? I have told you everything, now you need to just turn it in your head and figure out what is right induction statement?

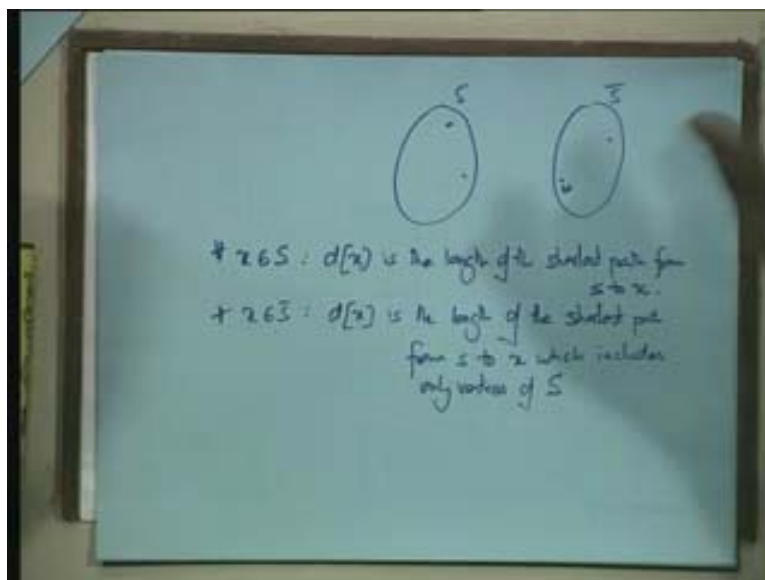
This is S and this is S complement (Refer Slide Time: 31:21). So the induction statement should be for all x in S . What should I write after this? $d[x]$ is the length of the shortest path from s to x . For all x in S complement, $d[x]$ is the length of the shortest path from s

to x which includes only vertices of capital S . (Hindi Conversation) induction hypothesis. How will this work? We will do the base case later.

We say there is a vertex w , which we move from here to here. Since it has moved from here to here, we have to argue for vertex w that $d[w]$ is the length of the shortest path from s to w . That we did here (Refer Slide Time: 33:26), $d[w]$ is the length of the shortest path from s to w . For the other vertices s complement we have to argue this statement that (Hindi Conversation) and that we have done in the slide which is given above that is $d[u]$ is the length of the shortest path from s to u that includes only vertices from s . Basically this was vertex u_1 , we looked at these three cases (Hindi Conversation) from s to u_1 which only include these vertices from s . That either includes the vertex w , if it does not include the vertex w then we know that the length of such a path cannot be no more than $d[u_1]$. If it includes the vertex w then if it were such a path then its length is no more than $d[u_1]$ but if it were such a path case two then its length is exactly this (Refer Slide Time: 34:36). So the minimum of these 2 quantities is the length of shortest path from s to u_1 which only includes vertices of $s+w$. (Hindi Conversation) for the other vertices basically sub vertices (Hindi Conversation) statement continues.

It will continue to remain true because we have not changed the distance labels at all. What have we changed as the result of this? (Refer Slide Time: 35:02-35:26) We have moved one vertex from here to here and changed the distance label of these vertices. Since we have changed the distance label of these vertices, we need to argue this statement of those vertices and since we move this vertex from here to here we need to argue this statement for these vertices and we have done that. (Hindi Conversation) which only includes vertices of s . No, you do not want to say (Hindi Conversation) if you just say that for all x , $d[x]$ is the length of the shortest path which includes only vertices of s . When w comes in here, you will have to look at all the vertices their and also argue that why w coming in, the length of their shortest path is not reduced.

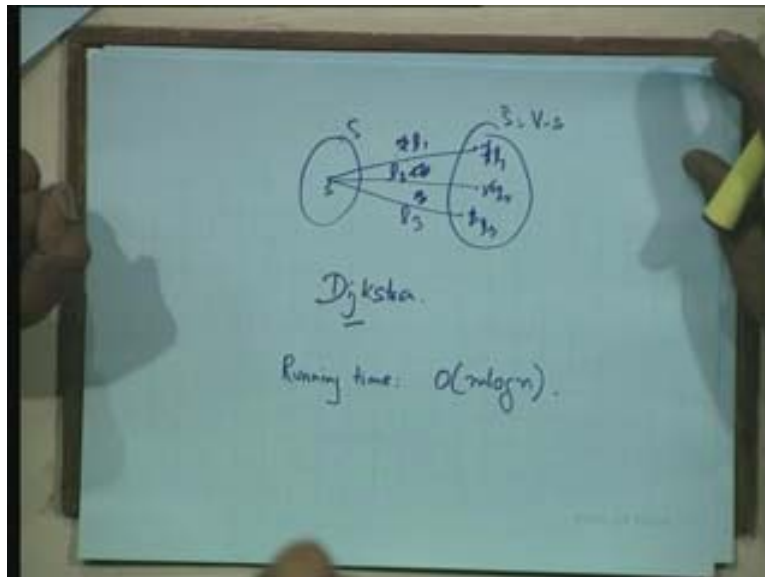
(Refer Slide Time: 36:13)



Because w only comes in now (Hindi Conversation) complicated. This is what exactly the thing is. So vertices (Hindi Conversation) number, right number and they will not change. That is the length of the shortest path from s to those vertices, period. And those vertices s complement (Hindi Conversation) that is qualified in a certain way. That is the length of the shortest path from s to that vertex, provided the path uses only the vertices of capital S . That is the qualification to the path (Hindi Conversation). This is the property that you have to keep in mind about these things. No, for the vertices which are already here, it is the same. We have not changed the distance labels for those vertices at all. If earlier it was true that for a vertex which was here, the d was the length of the shortest path from s to that vertex. Then it remains, it is the shortest path unqualified. This is not qualified (Hindi Conversation) it is the shortest path from s to that vertex.

Let us do the base case. What is the base case? But base case is very simple. The s has only little s in it and s complement is every thing but little s in it. (Hindi Conversation) basically length of this edge only. We start of by putting these as infinite but then the moment we move this across (Refer Slide Time: 39:00), this will get the length equal to this. So if this was 7, this was 12 and this was 3, then these would get distance labels 3, 12 and 7. Is this correct? Is this the length of the shortest path from s to this vertex which uses vertices of capital S only? Yes, there is no other vertex in capital S .

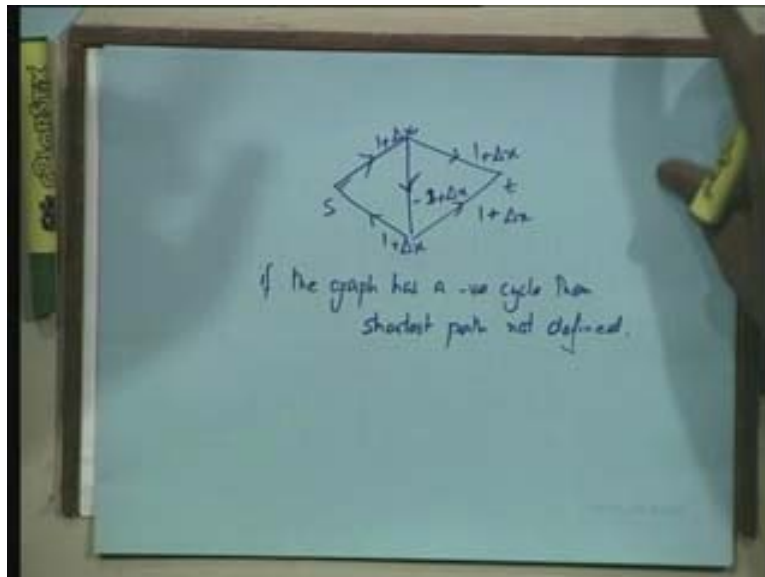
(Refer Slide Time: 41:30)



So there is only this one edge and it is the length of the shortest path. It is similarly for 7 and 12, so it is correct. The base case is correct and at each step we are maintaining these 2 properties. This is what you have to keep in mind. You just have to say this is length l_1 , this is length l_2 and this is length l_3 and this will have l_3 written on it, this have l_2 written on it and this will have length l_1 on it. And this need not be 3 edges and there can be any number of edges. This is a proof for any graph. (Hindi Conversation) that argues the correctness of this algorithm. This algorithm is due to Dijkstra where j is pronounced as y , so it becomes Dijkstra. What did we argue is the running time of this algorithm? This

algorithm would not work if the graph had edges of negative length. In fact if a graph has edges of negative length then actually there is no notion of a shortest path some times. Do you understand what I am trying to say? Suppose this was my graph, one comes repeatedly. So for instance the shortest path from s to t has length minus infinity. Why, because I start from here, go down, I come back and keep going (Refer Slide Time: 42:33). Every time I go around this cycle, I get minus one and I just keep doing this. So the problem is because you have a negative cycle. This is called a negative cycle. If the graph has a negative cycle then the shortest path is not defined. There is no notion of a shortest path any more, because the length of the shortest path could be minus infinity. But this graph need not have a negative cycle. For instance if this edge had a length of minus one, now what is the shortest path from s to t ? It is length one, I would go like this, come down and then go like this. What will be the length of this path? It is one. When do you think negative length make sense? Is this very artificial, negative lengths, can you think of a setting?

(Refer Slide Time: 47:25)



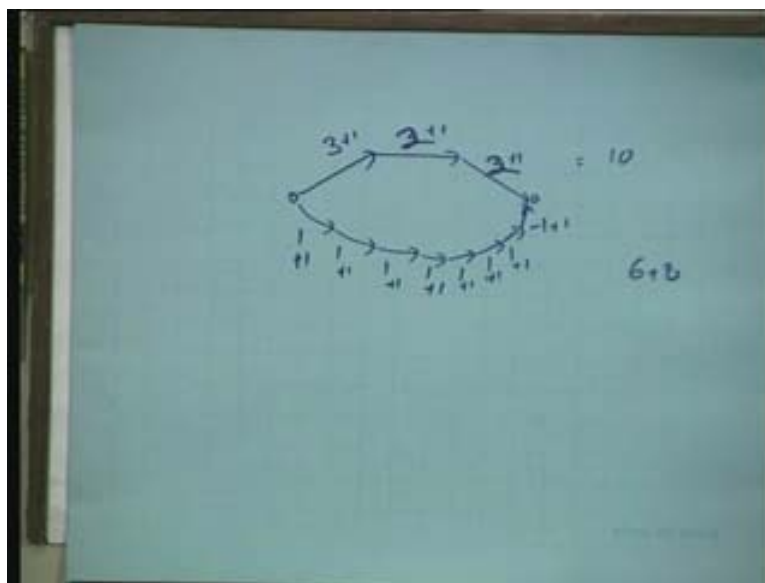
What other settings can you think where you can have negative edge lengths? So one other setting that I saw somewhere was, this is a graph which represents currency tradings. You can think of it in a different setting also. So each node is a currency, you are a global currency trader. And then an edge reflects that if I change from, if this is let say rupees, this is **Indonesian path** then this is the profit I incur in doing that. And negative would then correspond to a loss because of what ever I exchanged at that point of time.

And then finally you want to change your rupees into, let say back into rupees. What you are seeing if there is a positive cycle in this graph. **Then you will just keep going around the cycle and keep making when see** (Hindi Conversation) I do not understand your example but there are many such setting where negative would make sense. How can I solve this or how can I find out the shortest path in graph which has negative edge

lengths? Does Dijkstra's algorithm work here? Keep the minimum as zero and shift increase everything by plus one that is by the minimum (Hindi Conversation). Add sufficiently a larger number so that there is no negative length. Brilliant idea except that it does not work (Hindi Conversation). You add a delta x so that everything becomes positive. Why does this not work? Number of edges on one path could be different from number of edges on another path. Suppose I have a very simple graph (Hindi Conversation). What am I trying to show you? What will I do? Let me finish this example first. So this path is the shorter path but its length increases because there are a lot of edges on it. Let say all edges had length one on it and one edge had a length of minus one. So what is the length of this path?

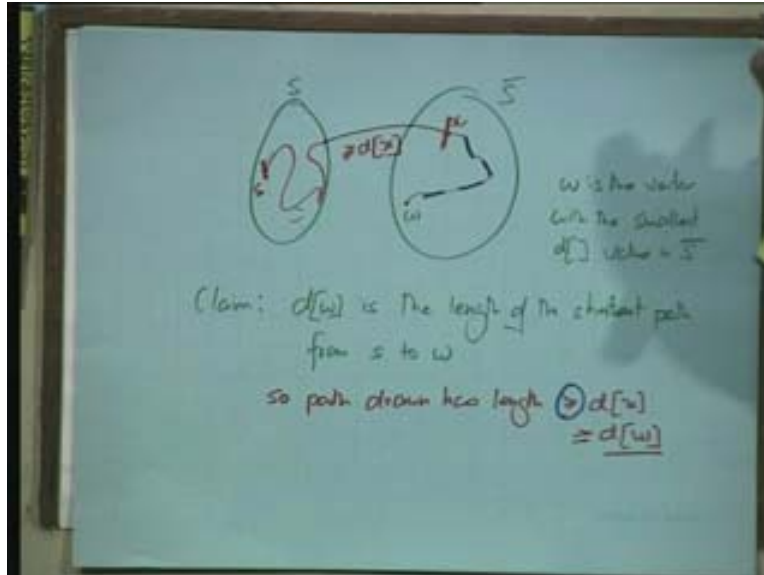
The length of this path is $3+4 = 7-1=6$. Let us make this path of length longer than 6, so let make it 3, 3, 3. This becomes 9. If I increase everything or let me make it 3, 2, 2 just to be on the safe side. So it is 7, longer than this. So this is the shortest path but if I now increase every edge by one, then what is the length of this path? (Hindi Conversation). So (Hindi Conversation) what do you mean reconvert back? Keep track of number of edges. How will you keep track of number of edges on the path? (Hindi Conversation) There is no notion of a shortest path if there is a negative cycle. But the graph need not have a negative cycle, let say this graph there is no negative cycle in the graph. Negative cycle (Hindi Conversation) is well defined.

(Refer Slide Time: 49:55)



So first tell me what goes wrong in the dijkstra's algorithm? Where does it break down? Did this argument break down? Here we argued the $d[u]$ is the length of the shortest path from s to u that includes only vertices from s . Does this break down? (Hindi Conversation) This part of argument is fine I think (Refer Slide Time: 51:10). But this part of the argument, when we said the $d[w]$ is the length of the shortest path from s to w , is not correct any more. Because I could have a path which goes like this. But this is shorter than $d[w]$ because there are plenty of negative edges on this part.

(Refer Slide Time: 51:43)



So this inequality is not true. (Hindi Conversation) Now I cannot say anymore that $d[w]$ is the length of the shortest path from s to w , this thing breaks down. So dijkstra's algorithm does not work, you cannot use dijkstra's algorithm to compute shortest path if you have negative edge length. (Hindi Conversation) How are you going to make your millions? How will you do that tell me?

So this is a valid question here. We will basically not discuss it today, because we do not have the time do it. So that is one thing remains to be done. (Hindi Conversation) Every vertex has linked list of edges. What does it do with the linked list of edges? What way does it maintain a pointer to? (Hindi Conversation) The shortest path from s to w and then the edge $w u_1$. (Hindi Conversation) and so on (Hindi Conversation).

We are going to look at it in more detail in the next class. How exactly you are going to do it, what is the modification to the code that will be required do such a thing. That is the one thing I would like to cover. The other thing I would like to cover is also to show you an algorithm to compute shortest paths, when you have negative edge lengths. But it is going to be very different from this algorithm. These are the two things that remain, we will take this up in the next class.