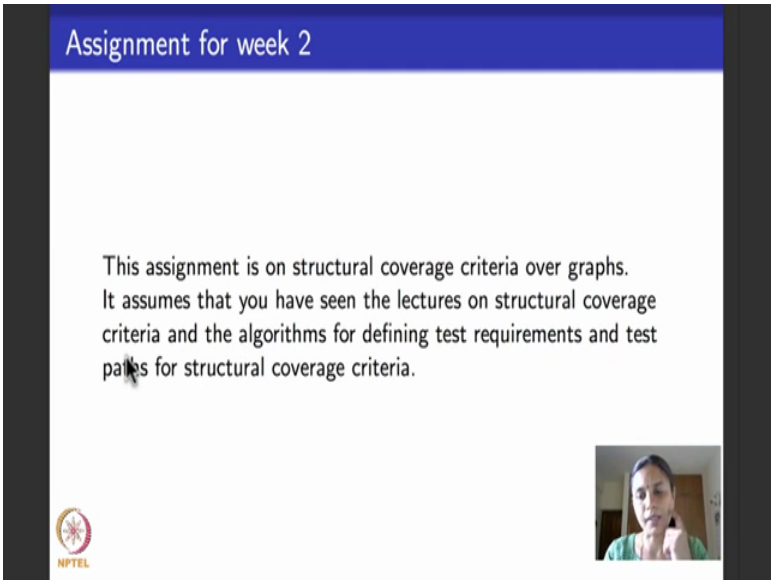
Software Testing
Prof. Meenakshi D'Souza
Department of Computer Science and Engineering
International Institute of Information Technology, Bangalore

Lecture – 10
Assignment 2: Structural Coverage Criteria

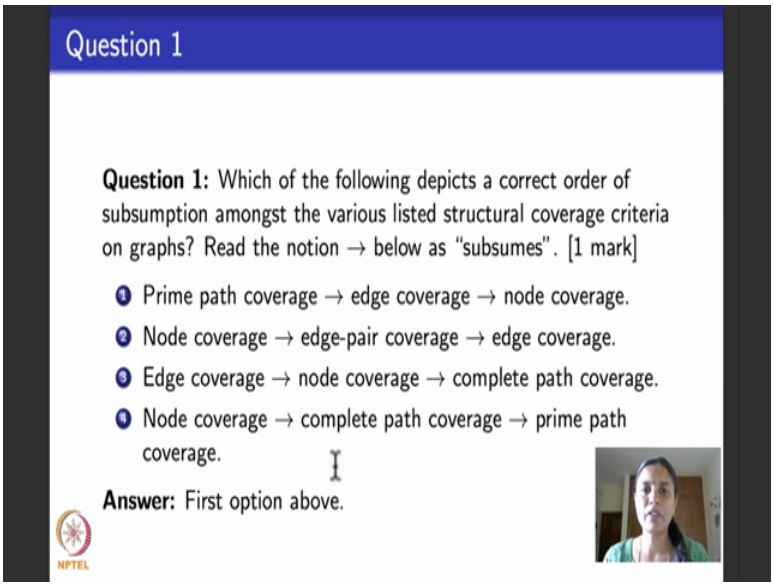
Hello everyone. So, this is the first lecture of third week. What I wanted to do today that is not really a lecture, but more to help you walk through the second week's assignment, assuming that you have looked at it and you made an attempt to solve it and you have also uploaded solutions. We will see how we will go about solving it, what are the questions, let us discuss our understanding of structure and coverage criteria over graphs to see if we could solve this assignment fully, correctly.

(Refer Slide Time: 00:43)



This lecture assumes that you have gone through all the videos of the graph structural coverage criteria, the algorithms for graph structural coverage criteria, and you have also looked at the assignment for the second week and made an attempt to solve it. So, what I will do now is while walk you through one question after the other, tell you what the a correct answer is and tell you how to get the correct answer.

(Refer Slide Time: 01:06)

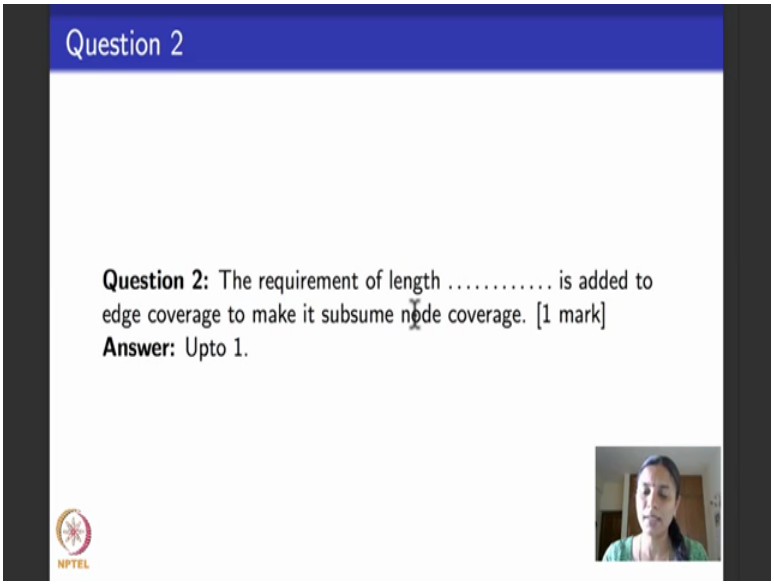


So, which was the first question the assignment had 8 questions, the first question talked about coverage criterion subsumption; to read out the question it asks which of the following depicts a correct order of subsumption amongst the various listed coverage criteria below. So, this notion, this symbol right arrow you should read it as subsumes. So, how do I read the first option? Your answer is to choose one of these options the correct option. How do I read the first option you read it as prime path coverage subsumes edge coverage which in turn subsumes node coverage, similarly second one would be node coverage subsumes edge pair coverage which in turn subsumes edge coverage and so on? So, the correct choice for this answer for this question happens to be the first answer, that is this on a prime path coverage that subsume edge coverage which in turn subsumes node coverage.

Now, why are the rest of them not the correct option if you see the second option it says node coverage subsumes edge pair coverage which in terms subsumes edge coverage. So, it as to subsumption relation like all other options this the second one edge pairs coverage subsuming edge coverage is correct, but node coverage does not subsume edge pair coverage. So, this is not the correct option similarly for the third one edge coverage does subsume node coverage, but node coverage clearly does not subsume complete path coverage. In fact, complete path coverage could even be infeasible as we discussed several times.

So, third one also cannot be the right option. What is the fourth one it says node coverage subsumes complete path coverage which in terms of subsumes prime path coverage. So, this parts correct, complete path coverage that subsumes guide path coverage because if I do complete paths I always include prime paths in (Refer Time: 03:01), but this is not correct as I told you in the option for the previous answer node coverage does not subsume complete path coverage. So, except for 1 the other options 2 3 and 4m 1 of the listed subsumption is not correct. So, none of them can be a correct answer in the first one both the listed subsumption criteria happened to be correct. So, the first one is the correct answer.

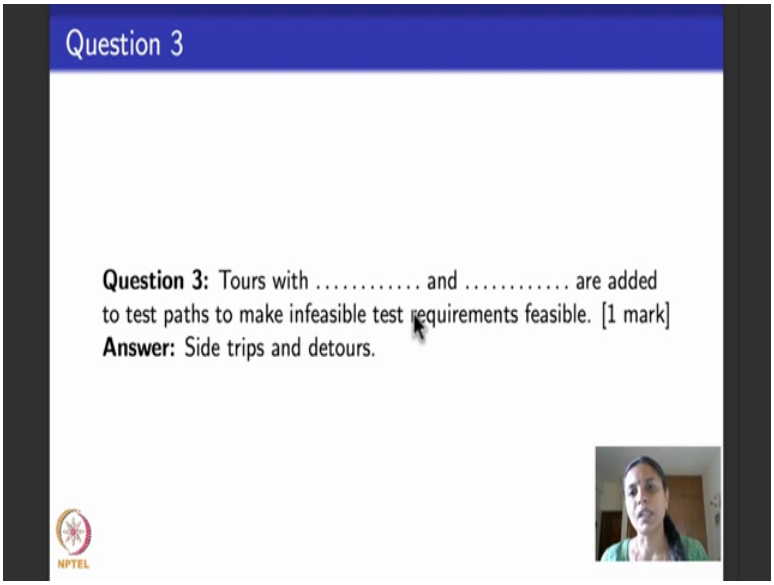
(Refer Slide Time: 03:28)



Now, moving on to the second question second question was simple fill in the blank question, it asks the following it says the requirement of length dash should be added to edge coverage to make it subsume node coverage. The answer is requirement of length up to 1, if you remember what edge coverage is the test requirement or t r for edge coverage say you cover all the edges of the graph. So, all the edges of the graph means all paths of length exactly 1. So, I do not keep it as exactly one I make it as a length up to 1 because I want edge coverage to subsume node coverage. Node coverage t r says nodes the path of length 0. So, if I include the requirement length up to 1 it includes paths of length 0 and it also includes paths of length what. So, because I want edge coverage to subsume node coverage, I change the requirement for edge coverage to be all parts of

length up to 1 as it is test requirement. So, the answer for this would be all paths of length up to 1.

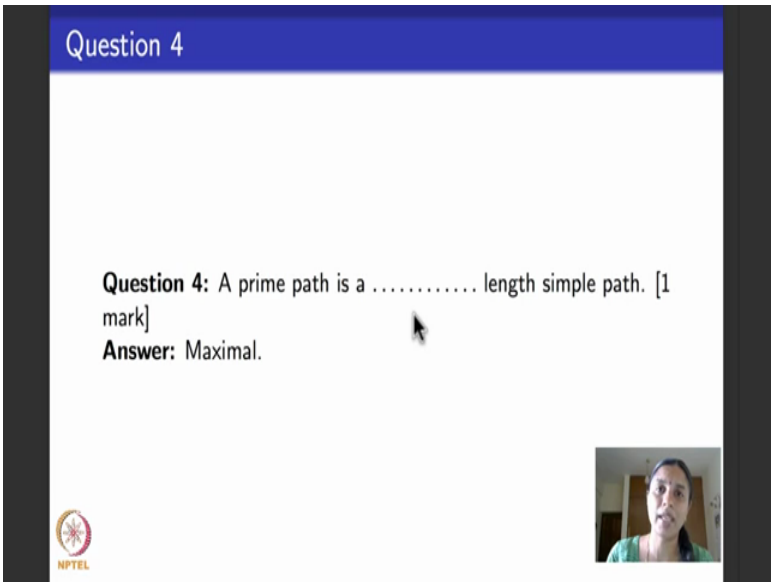
(Refer Slide Time: 04:31)



The next question asks again a fill in the blank question it says tours with the dash and dash are added to test parts to make infeasible requirements feasible right. If you remember the lectures we have discussed about side trips and detours. Side trip basically you take the test paths and then at some point in a vertices test path you decide to move out take another small side trip come back and join the vertices right. What would be a detour, detour is very similar to a side trip, but it might skip some edges also on the side trip. Why would I need this if you remember test path especially those dealing with prime path coverage could sometimes become infeasible, the test requirements will become infeasible.

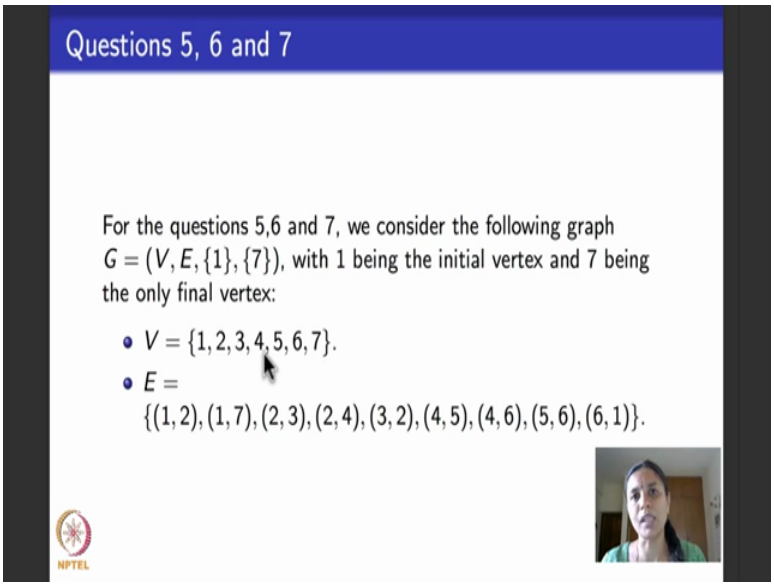
Especially if I have a loop like do while loop and I look for prime path coverage on that do while loop and may not be able to skip the loop at all right because do while says you first do one execution of the loop and then you check for the condition. Whereas prime path coverage will insist skipping the loop. So, I say I meet that test requirement for prime path coverage by using a test path that will skip the loop, but actually execute the loop as a side trip right. So, they are very useful to make infeasible test paths as feasible.

(Refer Slide Time: 05:56)



So, the next question was again fill in the blank question it asks what sort of path a prime path is the prime path, if you remember the definition is a simple path there does not come as a sub path of any other path. So, prime path a water called maximal simple path, maximal length simple path. Why do I put maximal and not maximum? I hope you know the difference maximal says that they could be more than one maximum length simple path right. There could be you see the examples if you remember that we looked at while dealing with algorithms for structuring coverage criteria there is prime path of length 2 they were prime paths of length 3 and there were prime paths of length 4. For each of these lengths that happens to be the maximum length simple path. So, that is why we say maximal. So, the answer to this in a prime path is a maximal length simple path.

(Refer Slide Time: 06:48)

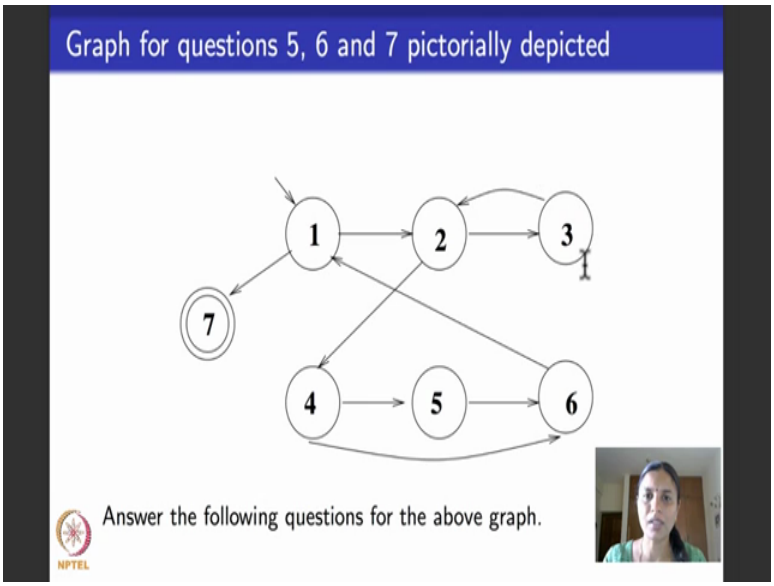


The last 3 questions in the assignment where to deal with coverage criteria that is based on an example graph that was given as a part of the assignment to you. So, here is a graph. So, the graph had 7 vertexes. Numbered 1 to 7, the vertex 1 was the initial vortexes start vertex the vertex 7 was the only final vertex and these were all the edges in the graph. And then you had 3 questions 5, 6 and 7 which dealt with answering questions about coverage criteria on this graph.

So, before we go ahead and look at how to answer the questions 5, 6 and 7, whenever I am asked a question like this the first wise thing to do is to be able to draw the graphs to be able to visualize the graph for yourself. What are the benefit is of drawing the graph, one is the notion of how the edges are placed what the edges are and how to look for paths in the graph becomes visually and it becomes a little more clear right? Then instead of staring at the set repeatedly to figure out what the notion of path is and things like how to achieve no coverage how to achieve edge coverage even to the extent of how to achieve prime paths coverage can be looked at the visual graph and answered very easily intuitively you do not even have to use the algorithms, that we discussed to be able to come up with test requirements and test paths for these criteria.

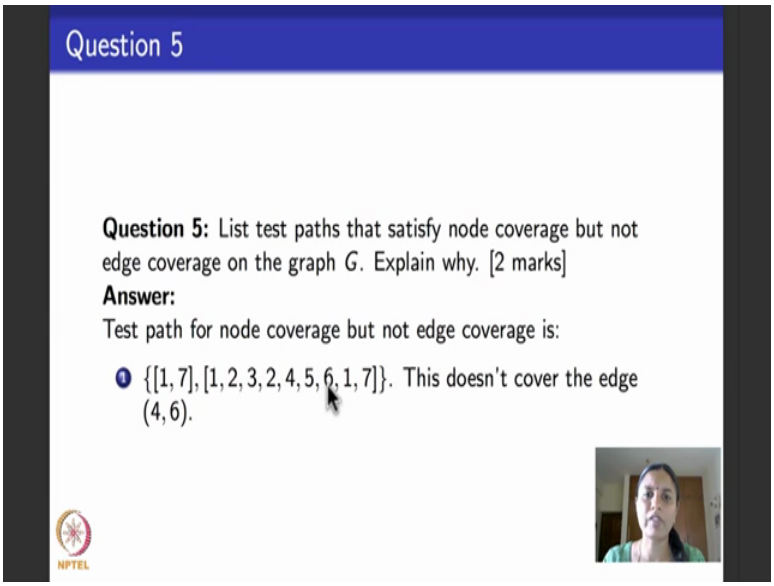
So, that is what I have done for you. Now I have drawn this graph. So, this graph has 7 vertices 1, 2, 3 and so on up to 7.

(Refer Slide Time: 08:17)



So, I have put this 7 vertices another thing to note is that while drawing the graph you may not get the layout of the graph. So, perfectly fine do not worry about that if you have to draw long edge short edge queued edges you still it is still helps to have a visual graph. And then a mark this initial vertex one mark the final vertex 7 using double circles, and then I have marked the edges if you see there is. So, many edges about 9 edges. So, there were 2 edges going out of 1 1 1 2 and 1 to 7, I have mark those edges 1 to 2, 1 to 7, 2 edges is going off 2, 2 to 3, 2 to 4, marked here 2 to 3, 2 to 4 only 1 edge going out 3 which is from 3 to 2, 3 to 2 to edges going on 4, 1 to 5, 1 to 6, 4 to 5, 4 to 6 one edge going on a 5 to 6 and 1 edge going from 6 to 1. So, here is one edge going from 5 to 6 one edge going from 6 2. So, I think we got captured all the edges here. So, this is the graph that is the same as this, but wrong.

(Refer Slide Time: 09:31)



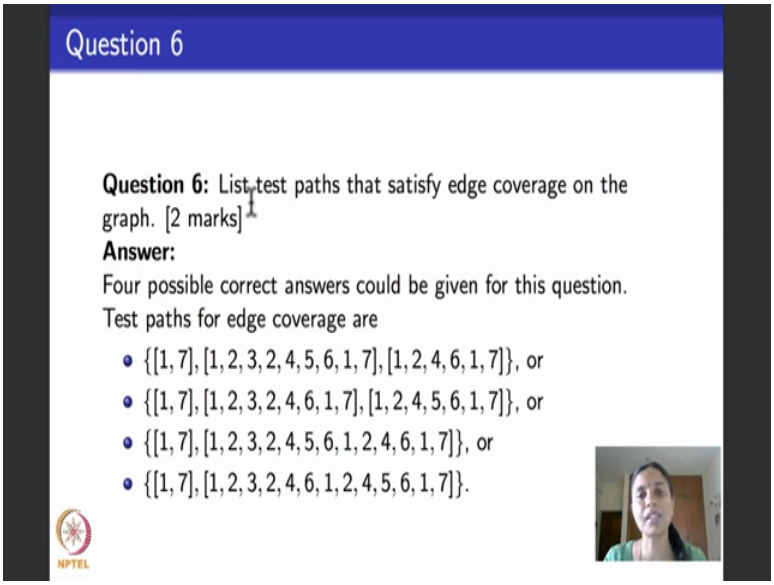
So, now let us go ahead and look at the questions that were asked about this graphs. So, question number 5 says to list all the test paths that satisfy node coverage, but it also says that some of list tests path in such a way that that they satisfy a note coverage, but do not meet edge coverage if you remember node coverage does not subsume edge coverage in all cases. So, in this particular graph node coverage need not subsume edge coverage. So, it might be possible to come up with a set of test path that satisfy node coverage, but not edge coverage. So, what is node coverage test requirement for this, graph it is basically the set consisting of all the 7 vertices what are the test paths for node coverage if you see in this path from 1 to 7, is like a lone of path right it is on it is own. So, you have to be able to take this this is the only way to go to 7. So, to do node coverage for the vertex 7, I need to be able to include this as a standalone test path. It is a test path because it satisfies all the requirements of a test path it begins at an initial vertex which is one it ends in the final vertex which is 7.

So, I have to be able to include this test path to node cover 7. So, the remaining nodes are 2 3 4 5 and 6. So, how do I cover them I can include any test path remember test paths always have to begin in in initial vertex and end in a final vertex. So, I begin at one I do not have much of a choice, here I go 2 from 2; let us say I go to 4, 5, 6 from 6; I go to 1. So, which are the nodes I have covered if I had taken such a path I have covered 1, 2, 4, 5, 6, earlier I had covered 1 and 7 and left on 3; right, but I have not still not left out. So, I can always go back and then I will continue this path that take it from 1 to 2, I visit to

once again that is all right there is no harm in dash because test paths can have cycles, and then I go to 3. So, I have covered now all the vertices, but I cannot end my test path here because the test path has to end in a final vertex. So, I take this vertex back, and now I still cannot go to 7 the only way to go to 7 is to go to 4 and then may be do this 4 to 6, 6 to 1 and 1 receive.

So, basically what I am trying to say is that if my test requirement is node coverage then they has to be a test path that consists of this segment in the graph which is just this edge 1 to 7. And for all the remaining vertices you need to go through this whole zigzag cycle once and come back to 7. It could include any test path the; that is that here, what I have done is I have included this test path I have d1, 1, 2, 3, 2, 4, 5, 6, 1, 7. So, to trace it out 1, 2, 3, 2, 4, 5, 6, 1, 7, it does need note coverage if you see all the nodes that are included in the test path, but I had one more condition in my question that was do not have do not make it subsume edge coverage. So, I am chosen test path in such a way that I have left out one edge which is this edge I have left out the edge 4, 6. So, if you see this test path achieves node coverage, but it has it almost achieves edge coverage it visits all the edges except for this edge 4, 6. So, this test path is good enough an answer for this question.

(Refer Slide Time: 12:53)



So, the next questions is list all the test paths that can satisfy edge coverage criteria on the graph. So, here they could be several possible correct answers. In fact, 4 different

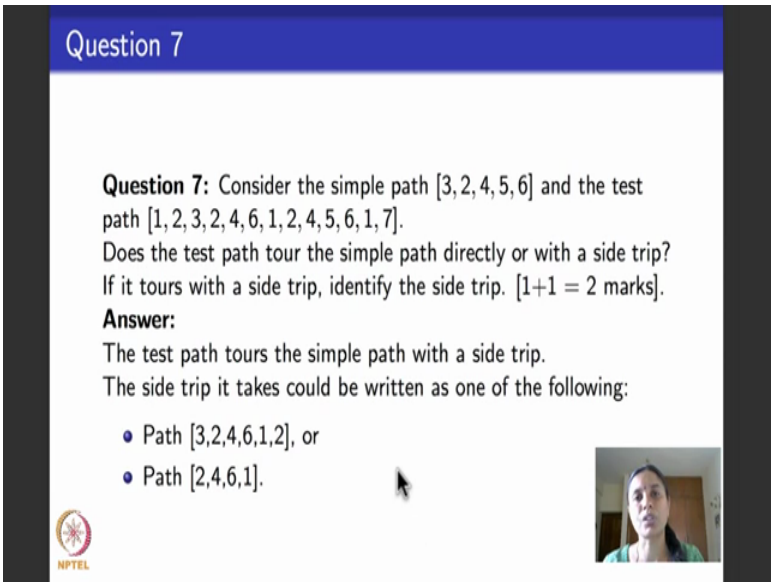
possible correct answers are there. Let us go back and look at the graph, as I told you right this path is the standalone path it always has to be taken whether you do more coverage edge coverage prime paths coverage. So, 1, 7 will always be there if you see in all the 4 options that I have listed here 1, 7 is all wasted and then there is a question of this covering the rest to the edges. So, if you see when I execute the only thing to remember is that all these are pretty much seeded it is only at this place do I have a branching or a choice of course, 2 also I have a branching, but it is this this, but once I do that again I have another choice here at 4.

So, suppose I take this (Refer Time: 13:45) branch 4 to 5 5 to 6. Then I have left out this one this inch 4 to 6. So, I need to be able to come back how do I come back only way to come back is to go from 6 to 1, 1 to 2, 2 to 3, 3 back to, 2 to 4 and then I have to take 4 to 6. So, the 4 options basically exploit this. So, you to do 1 7 separately, and what are these 4 options they let you do the let you traverse from 4 to 6 either using the 2 edges 4 to 5, 5 to 6 or using this single edge 4 to 6. So, it is 1, 2, 3, 2 use the edge 4, 5, 6, go back to 1 and then to 7 finish. And then you begin at 1 go to 2 use the edge 4 to 6 the direct edge do one and 7 again. This one is the same thing, but it says. So, I will just trace it out here you do 1, 2, 3, 4, 5, 6, 1, 7 for you do 1, 2, 3, 4, 6, 1, 7.

The second one what I have done is I have d1, 1, 2 4, 6, 1, 7 and then I have d1, 1, 2, 3, 2, 4, 5, 6, 1, 7. So, those are the 2 options here. And the last 2 options I have basically joined them both avoided the repetition of going back to 7 and beginning again at one I just go back to 1 and resume from 2 again. So, what I have done just to trace the path here is I have d1, 1, 2, 4, 5, 6, 1, 2, 3, 2, 4, 6, 1, 7 that is this one long test path that is here.

The second one reverses this order of visiting 4, 5, 6 and 4, 6. It does 4, 6 first and then 4, 5, 6. So, 4 answers for edge coverage do 1, 7 separately and then the rest of these parts visit them in any order that you like, but you have to be able to do this large strongly connected component twice, once going through this path and once going through this path and there is a choice here. Similarly to 1s if you exercise this then the next time you have to be able to exercise this that is the only way I can cover all the edges in the graph. So, that is the test path for edge coverage.

(Refer Slide Time: 16:02)



The last question of this assignment asks you to go back to the graph and consider the simple path 3, 2, 4, 5, 6. So, let us go and see what that simple path looks like with this 3, 2, 4, 5, 6. Please remember simple path is not a test path. So, it can begin an end at any node the only thing to note is that it is a simple path does not have any cycles.

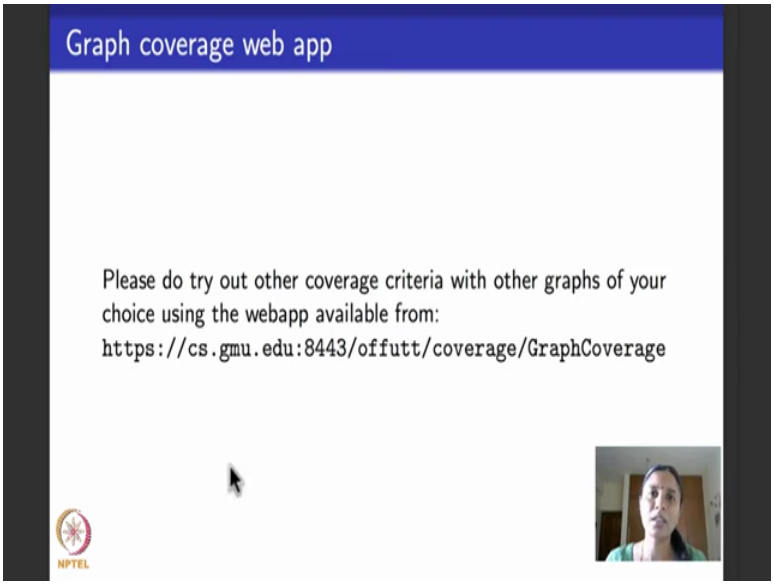
So, suppose you had a specified path coverage requirement of having to write a test path that covers this simple path. So, they have made an attempt given you some test path. So, the test path very long it has 1, 2, 3, 2, then that is the 4, 6 edge comes back to 1 and that is the 4, 5, 6 edge it comes back to 1 and then to 7. It is this long path the just walks through the entire graph once round and round. So, the question that was asked where to parts does this test path toward the simple path directly or with a side trip the test path does not toward simply path directly right, if you see the simple path is 3 to 4, 5, 6 it is split into 2 parts a 3 and 2 come here 4 and 5 and 6 come here.

So, it does not toward simple path directly, if it was the simple path directly the simple path will be a contiguous, sub path to the test path. Because it is not there it does not do other simply path directly. So, the test path in fact, towards the simple path with a side trip, what does the side trips side trip is this bit that is left in between. So, 4, 6, 1, 2, so, you could write it like this 2, 4, 6, 1 or you could include 3 and these 2 and make it 3, 4, 6, 1, 2. So, this test path towards this is given simply path with the side trip and the side

trip looks like this if you are written either of these it would be acceptable (Refer Time: 17:49).

So, I hope this mall exercise helped you to understand how to solve simple assignments that we would be giving every week. I will try to upload a couple of more videos as we move along for other assignments that will help you to solve these assignments.

(Refer Slide Time: 18:08)



For now, to get to know more about the graph structural coverage criteria I encourage you to look at the web app that comes as a part of this course book. It is a very nice app it will help you to learn how to represent graphs as adjacency lists, and how to feed them and how to do various coverage criteria. Please do try it out hope it be useful to you.

Thank you.