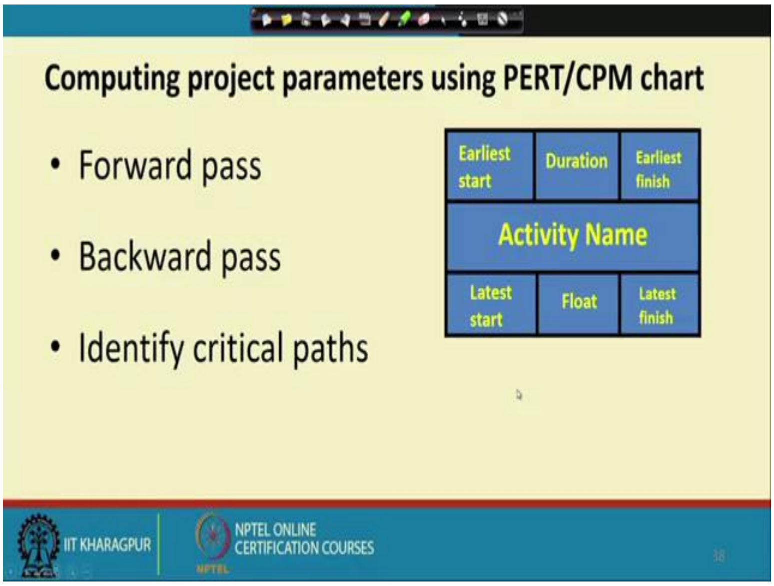
Software Project Management
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
Computation of Project Characteristics Using PERT/CPM: Illustration

Welcome to this lecture. In the last lecture we had looked at the PERT CPM technique by which we compute various project characteristics, the activity characteristics, the critical path and so on. Now, we will quickly review what we discussed in a minute and then we will take some examples to illustrate how to use the methodology. Very simple we can do it in a minute or two for moderate sized networks. Now, let us look at let us review first quickly and then get started with the example.

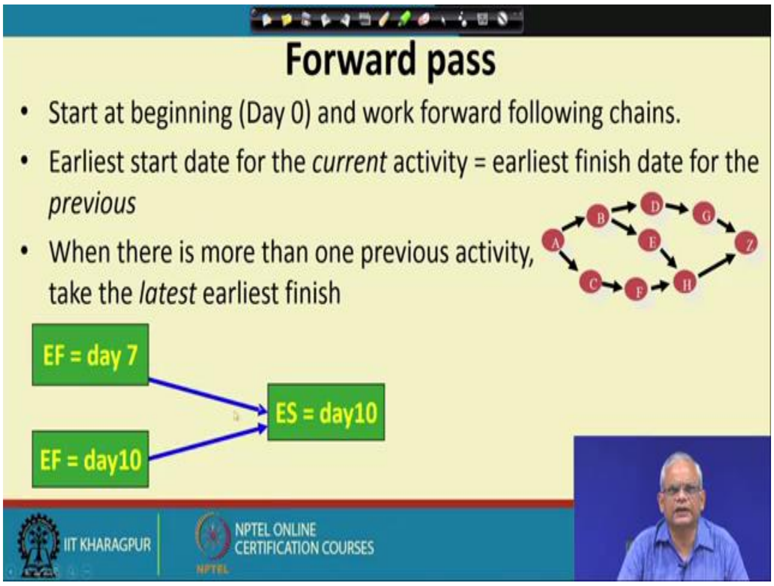
(Refer Slide Time: 01:09)



Just to recapitulate what we discussed in the last lecture, that we can compute the project parameters by using some simple methodology. We will first do a forward pass through the network; to start with we will have the activity names and the durations for all the activities and the precedence between the different activities in the form of a network. We will start with the leftmost activity or the start activity and then we will have a forward pass, where we will computer for every activity, the earliest start and the earliest finish.

In the next step we will do a backward pass, where we will start with the last activity and we will have the latest finish and the latest start computed. And, then we will compute the float which is the difference between the latest finish and the latest start. And, having done that we will identify the critical paths that is we will find out all the activities which have 0 float time. And, we will find a path through the network for which the float time is 0 and we will call it as the critical path we will mark it on red, these are of special importance to the project manager.

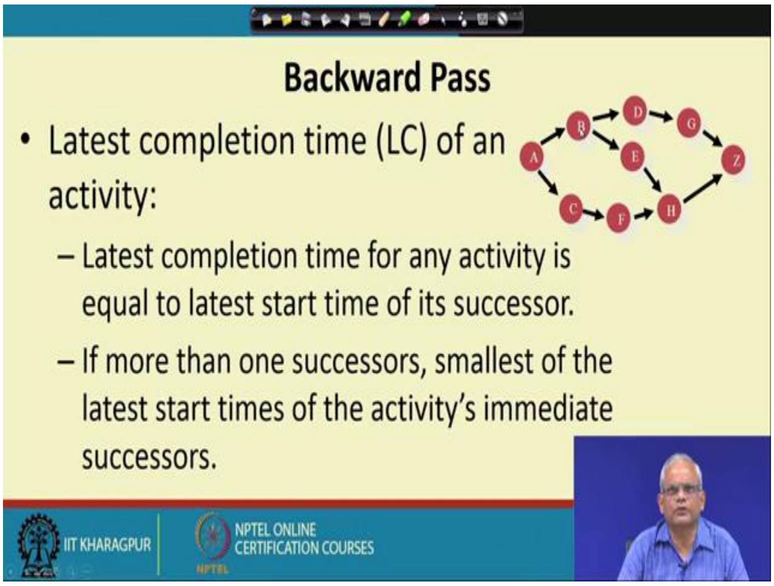
(Refer Slide Time: 02:52)



The forward pass we start from the beginning or the start task which is started day 0 and then we work forward. We will set the start date which is we will set it as the earliest finish date for the previous task. And, if there is more than one previous activity, we will take the latest finish time. There are more than one previous task and for the earliest start is the latest of the later of these two we will take 10 here.

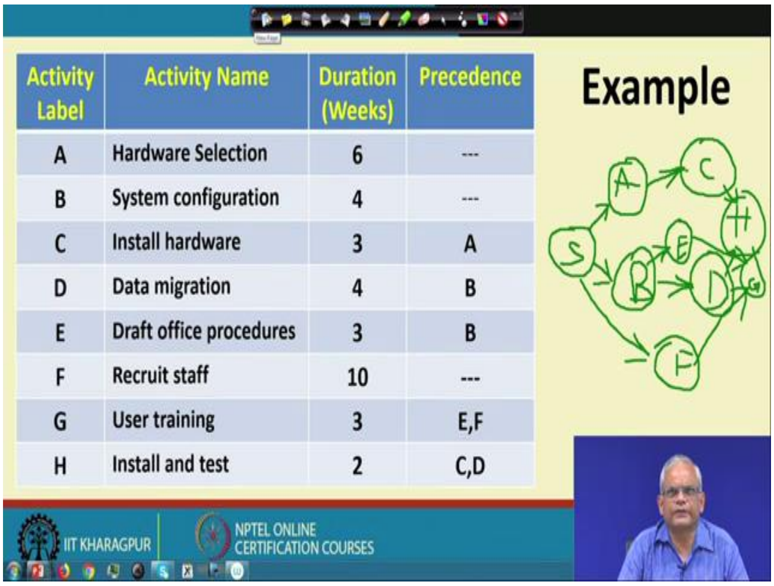
So, the earliest start we will set as earliest finish and then for the task we once we find the earliest start, we will add the duration to get the earliest finish for that particular task.

(Refer Slide Time: 03:53)



And, once having done that we will take the backward pass, start with the last task and we will compute the latest completion time which should be equal to the latest start time of its successor. The latest completion time of G should be equal to the latest start time of Z. When we take the example we will see that it is a very simple step, just a small arithmetic. The only case is that when there are more than one successor for a node, the latest completion time should be equal to the smallest of the latest start time of both the successors; this is the only thing that we must be careful.

(Refer Slide Time: 04:46)



Activity Label	Activity Name	Duration (Weeks)	Precedence
A	Hardware Selection	6	---
B	System configuration	4	---
C	Install hardware	3	A
D	Data migration	4	B
E	Draft office procedures	3	B
F	Recruit staff	10	---
G	User training	3	E,F
H	Install and test	2	C,D

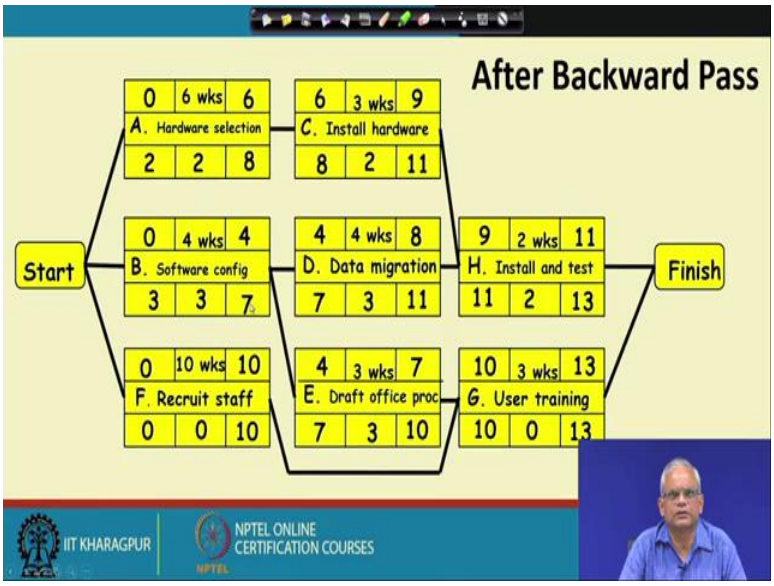
Now, let us take an example: here we have project where we have several activities which have been identified. Hardware selection, system configuration, install hardware, data migration, draft office procedures, recruit staff, user training and install and test. In our task network it will be difficult to write all these levels and the activity nodes and therefore, we use the label A B C D E F G H. And, the duration in weeks written here this is the estimated duration of every task. These are deterministic times we know well, but later we will see that how to use probabilistic estimates.

The precedence is also identified A B have no precedence and C has precedence of A, if we draw this A has no precedence, B has no precedence. We just draw it as it is, we have for C A is the precedence, we draw C and draw an arrow between A and C to show the precedence. And, we have for D B is the precedence draw an arrow between B and D, showing that B is the preceding activity for D until B completes D cannot start. And, since we have only we should have only a single start node which will be a day 0 we will just write as start node here, we cannot have two start nodes A and B.

So, even though there is no dependency for A and B you just draw a start node, just to have a single start node and then for E we will have B as the precedence. So, we will draw here E and for this B will be the precedence and then we will have F for which there is no precedence. And therefore, S will be its precedence and then for G both E and F are precedence. So, we will draw G here and for which both E and F are the precedence and then H for which C and D are the precedence. For H C and D are the precedence, you can draw it nicely because here there are arrows intersecting should not normally have that. And, then for H and G there are two end nodes, we will put another E node which is the successor node.

Now, once having drawn that we will apply our forward pass and backward pass.

(Refer Slide Time: 09:25)



So, this is the one, if we redraw the one that were drawn in the previous slide we will have this. I have also written down here the names of the activities, but that is not necessary A B C D etcetera should be enough. So, A B and F these are the successor node at the start node, start and finish are dummy nodes, duration is 0, start date of start task is day 0. And, then A takes 6 weeks, B takes 4 weeks, F takes 10 weeks that we can get from the table that has been given and for every task we write the name of the task and also the duration.

Now, let us apply the forward pass, in the forward pass the start task starts and completes on day 0 and therefore, this can start on day 0. We will put 0 here and then this will be 6, the earliest start time is 0, the earliest finish time is 6. Similarly, this is earliest start time is 0 that is end of day 0 or start of day 1 and we put 0 here as the earliest start date because, this has completes day 0 and then we add 6 here put 6 here, 4 here and 10 here. So, that we have done here that for all these 3 tasks A B and F, we in the forward pass we first the completion time of this is 0 0 0.

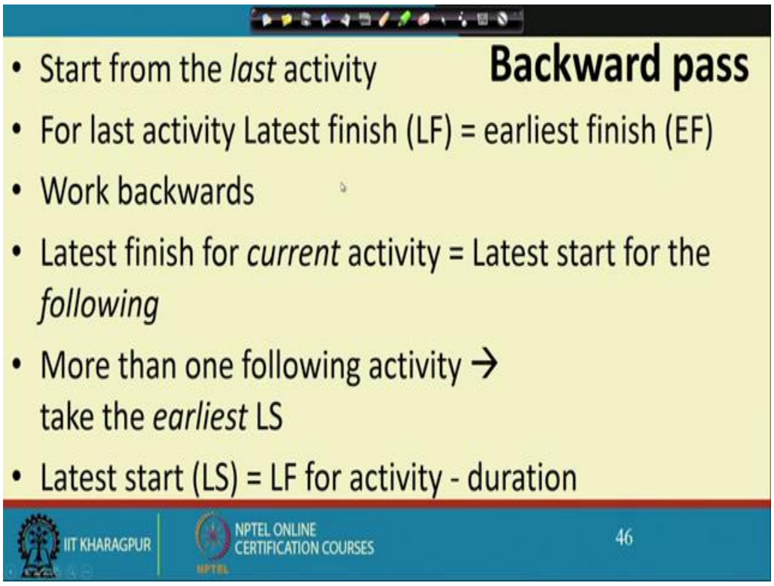
And, add the duration here 6 and put 6 4 and 10 very simple arithmetic and in the next step we look at the successor here C is the successor for A and we put 6 here indicating that. The earliest completion time of task A is 6 and therefore, the earliest start time of task C in 6. And, then it takes 3 weeks we put 9 here that is the earliest completion time.

For task D the earliest start time will be 4 and earliest completion time is 8 and for task E the earliest start time is 4, earliest completion time is 7.

Now, we have just written that the earliest time start time is 6, completion time is 9 added 6 plus 3 is 9 and here earliest start time is 4, earliest completion time is 8. And, here earliest start time is 4 and earliest completion time is 7. In the next step, we will compute for H, but for H just see that it has two predecessors D and C and both have to complete for H to start. Even though D complete at 8th day end of 8th day, H cannot start on 8th; it has to start on 9th that is end of 9th day. Now, G also has two predecessors that is F and E and even though E completes at 7, it cannot start on 7 it has to start on 10.

We will put here 9 and 10 and this becomes 11 and this becomes 13. So, I have done that so, that is the end of the forward pass, all the earliest start time and earliest completion time are all marked on the tasks. Now, we will start the backward pass where we will complete the latest start time and latest finish time for all the tasks. We will start from the finish node, the finish node, the latest finish time is 13 that is both have to complete; even the earliest finish time is 13 because both have to complete. So, for the last node the earliest finish time is equal to the latest finish time which is equal to 13.

(Refer Slide Time: 15:08)



And, we will start the backward pass for the last activity we will set the latest finish time is the earliest finish time as I was mentioning from then we work backwards. The latest finish time for a task will be the latest start time of the following task and either more than one following activity we will take the earliest start time. And, then from the latest start time, the latest finish time we will compute the latest start time by subtracting the duration.

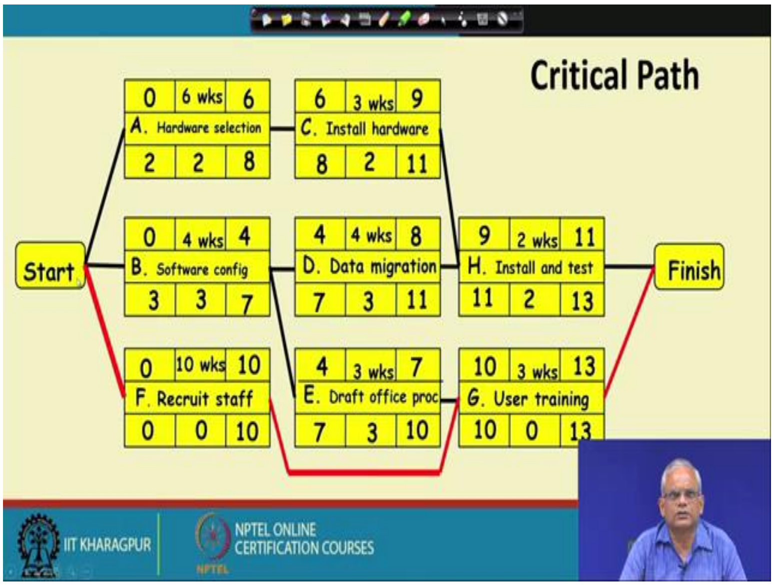
Now, let us start the backward pass the finish here it is 13 ; that means, that both of these have to complete by 13, they have single successor. So, the thing is very simple we just write 13 and 13 here and then we have the duration is 2 weeks and we from 13 we get that here it must be 10. And, here it must be 11 subtract 2 from here sorry 2 from here we get 11 here and from 13 we get 10 here. Now, in the next thing that we compute is the slack time which is the difference between the earliest finish and the latest finish; sorry the latest finish mean earliest finish or the latest start minus earlier start both will give 2.

So, we put 2 here, we put 0 here; now let us computer for this ones, now this has a single successor which is completing at 9 sorry 11. And therefore, the latest it must complete by 11, similarly for this and for this it is 10 and we must have 10 here. So, 11 11 and 10 we have written here and then we subtracted 3 from here and got 7 and 7 minus 4 or 10 minus 7 is 3 and similarly here 11 minus 4 is 7 here and here 11 minus 3 is 8 and 8 minus 6 is 2. In the next step for this it is simple, the preceding the succeeding task for this F is G and G the completion the earliest start, sorry the latest start time is 10.

And therefore, it must complete by 10 and we put 10 here. But what about B? B has two tasks here a successor D and E, but then for both of this it is 7 so, we can put 7 here and for this we have C as the successor for A, C is the successor and 8 is the latest start time. So, we will put latest completion time for A as 8, let us do that here both were same and that is why you put 7 here. If one was 8 and one was 7 we would have put 7 here, if one was 8 and 6, we would have put 6 here and then 7 minus 4 is 3 and 3 minus 0 is 3 is the slack time.

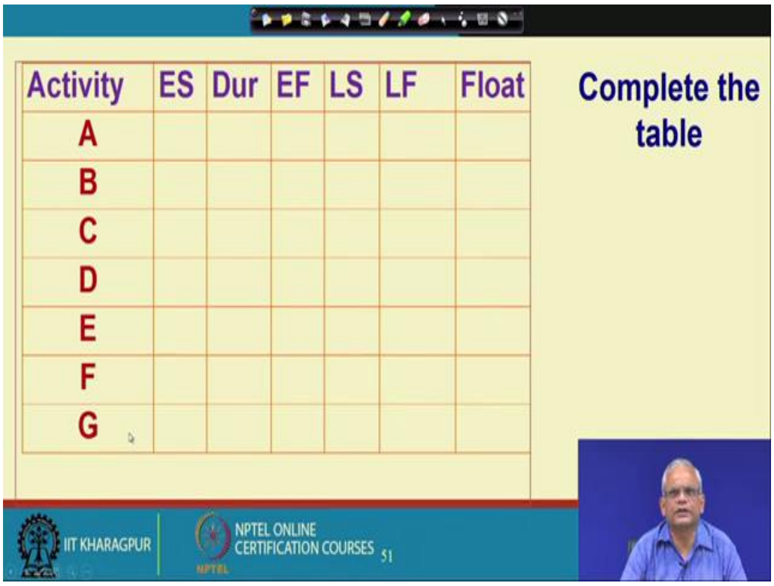
Similarly, here 10 minus 10 is 0 and 0 minus 0 is 0, 0 slack time and for this similarly 8 minus 6 is 2, we write 2 here and then the slack time is 2 minus 0 put here 2. And, we have got all the task characteristics here, we have completed the backward pass. The next task is to identify the critical path, let us find out all the tasks that have slack time 0; here we find that F has slack time 0 and we have G has the slack time 0. So, we indicate this path as the critical path we will draw it using red colour ok.

(Refer Slide Time: 20:17)



We drew it using red colour. So, this is the critical path, critical path identification becomes very simple once we identify all the activity characteristics and we can write this in the form of a table all the task parameters.

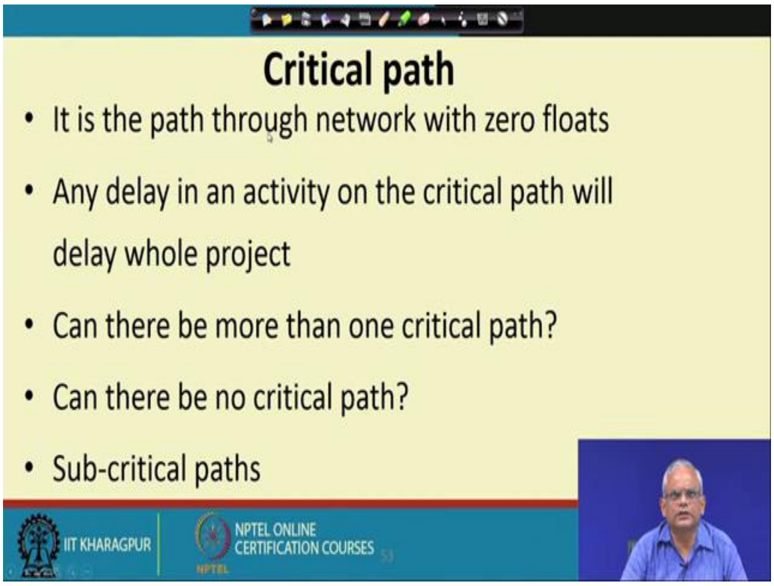
(Refer Slide Time: 20:49)



Activity	ES	Dur	EF	LS	LF	Float
A						
B						
C						
D						
E						
F						
G						

We can complete this table; we have A B C D E F G as the tasks, the earliest start time, duration, earliest finish time, latest start time, latest float time sorry, latest finish time and the float time. We can even write in the form of a table, we will just copy from the diagram; sometimes this representation also becomes useful to the project manager.

(Refer Slide Time: 21:32)



The critical activities here are the activity F and G, the critical path as we saw its the path through the network with zero float. It is very important for the project manager to pay careful attention to the critical activities because, any delay on a critical activity will delay the whole project. But, just out of curiosity some questions one is that: can there be more than one critical path for a project? The one example we discussed, we had only one critical path that is the slack time were 0 on one path and we identified that as a critical path.

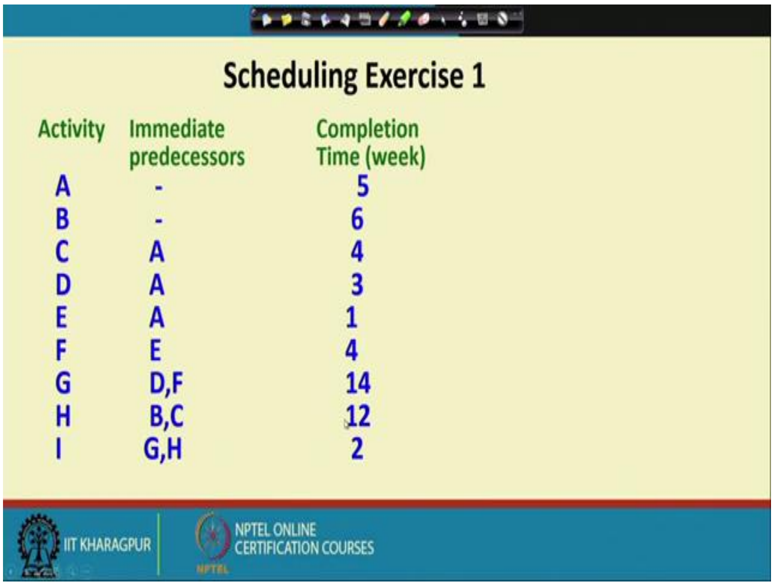
But it may so, happen especially for larger task networks that there is more than one critical path, it is possible that more than one critical paths and on each path the slack time available to the activities is 0. But, is it possible that there is no critical path for a task network? The way we computed there has to be a critical path, the forward pass and backward pass. But, suppose we have a situation where the completion time is also predefined, the last task the latest completion time is predefined; it is not computed by the forward pass. And, then we can have it is possible that we can have no critical path, all tasks might have some slack time or other.

But what about subcritical paths? We identified all tasks or activities that have 0 slack time and we said that the project manager has to pay special attention to those tasks. But, what if all tasks have other than the critical path all tasks have like slack time of 10 7 etcetera, but there is one task which has slack time of 1. It is not 0, but slightly more than

0 and therefore, those tasks are also subcritical; that means, that we do not have too much laxity or too much flexibility with those tasks. The project manager also needs to consider the subcritical paths, where the activity is do not have too much of even though they do not have exactly 0 flexibility 0 slack time, but the slack time available is very less.

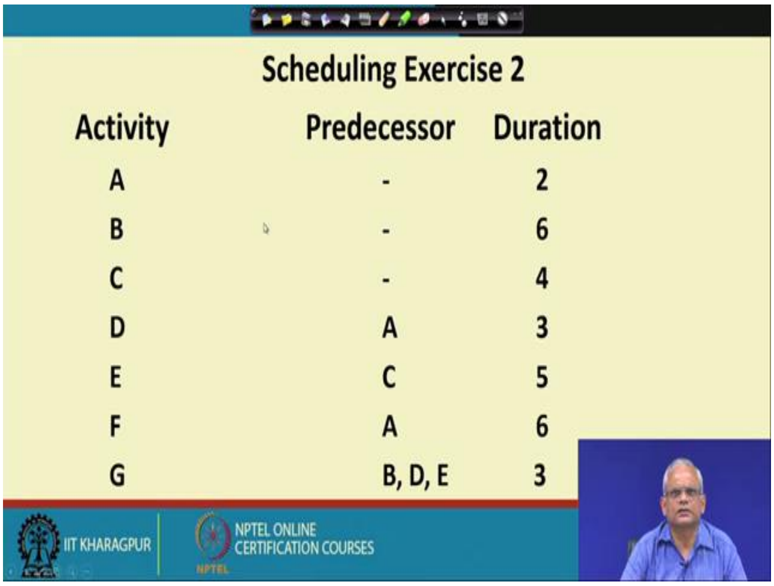
Especially for larger projects it may be also necessary to identify the subcritical paths, where the slack time even though is not 0 for the path, but then it may be a very small number.

(Refer Slide Time: 24:53)



Here are few exercise; please try this out, draw the task network and then use the PERT CPM methodology to identify the project characteristics.

(Refer Slide Time: 25:16)



There is another exercise, the activity name, activity label, predecessor of the activity and duration. Please try this out, identify the critical path, if not look at the example that we worked out, it is very simple. You have to use the forward pass, identify the earliest start time and completion times, then start the backward pass. And, you will have the latest start time and finish time and then compute the slack and based on that you can identify the critical path. We are almost at the end of this lecture; we will stop here and continue from the next lecture.

Thank you.