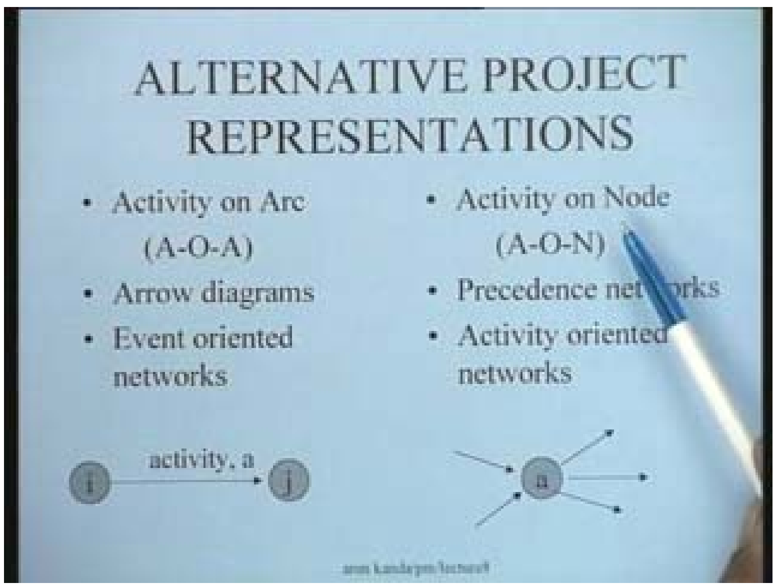
Project and Production Management
Prof. Arun Kanda
Department of Mechanical Engineering
Indian Institute of Technology, Delhi

Lecture - 9
Basic Scheduling with A-O-A Networks

Today we are going to be talking about basic scheduling on activity on arc networks. In fact the primary contribution of PERT and CPM has been to develop schedules for projects and this is in fact one of the core topics in using PERT and CPM. Before we go ahead we would like to talk about some alternative project representations. Projects can be broadly represented in 2 modes, the activity on arc mode and the activity on node mode.

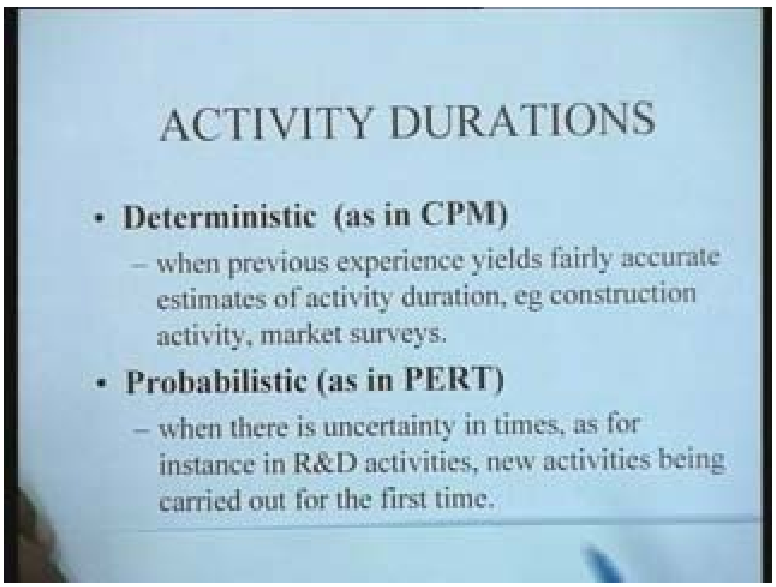
(Refer Slide Time: 1:46)



The activity on arc mode is popularly referred to as an arrow diagram also and in fact the these calculations are generally event oriented because in this A-O-A representation each activity is represented on the arrow and the starting node and the end node of each activity is given the interpretation of an event. It's a point of occurrence in time and therefore these algorithms which are there on A-O-A networks are often referred to as event oriented algorithms. On the contrary when you refer to an activity on node network these networks in common parlance are also known as precedence networks and the kind of algorithms which are used for scheduling here are typically referred to as activity oriented networks because each node on the network is now representing an activity and it is not representing a point in time or a milestone as it was in the A-O-A network.

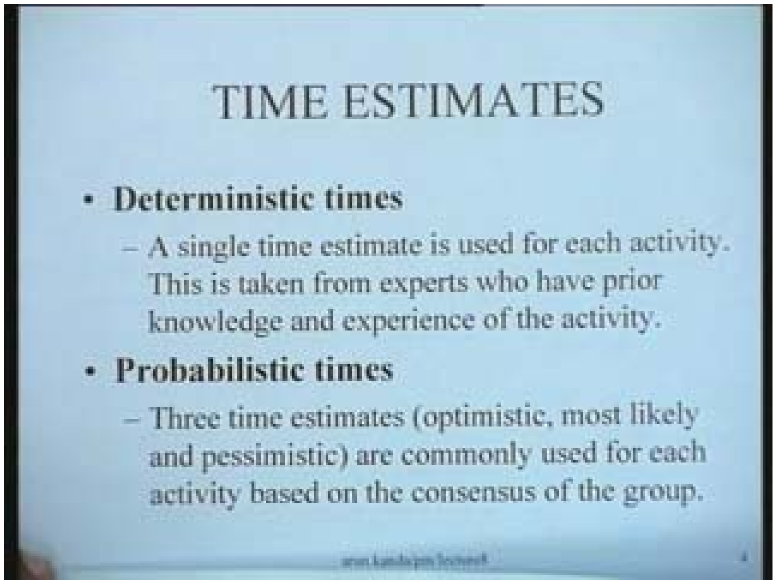
Then one has to be careful about the kind of representation that is used for various activities in the project network and activity durations can be either deterministic which is the case which is true in critical path methods or CPM and the important thing to note here is that when we are talking about deterministic estimates of time of activities these are used when previous experience yields fairly accurate estimates of activity duration and this would happen for instance in construction activity where you have estimates of time of various activities or in conducting market surveys for which you have previous experience and therefore you can estimate the times quite exactly. On the other hand we have probabilistic time estimates which happens for instance in PERT or in program, evaluation and review technique and the essence here is that there is considerable uncertainty in times as for instance in research and development activities or when new activities are being carried out for the first time.

(Refer Slide Time: 4:29)



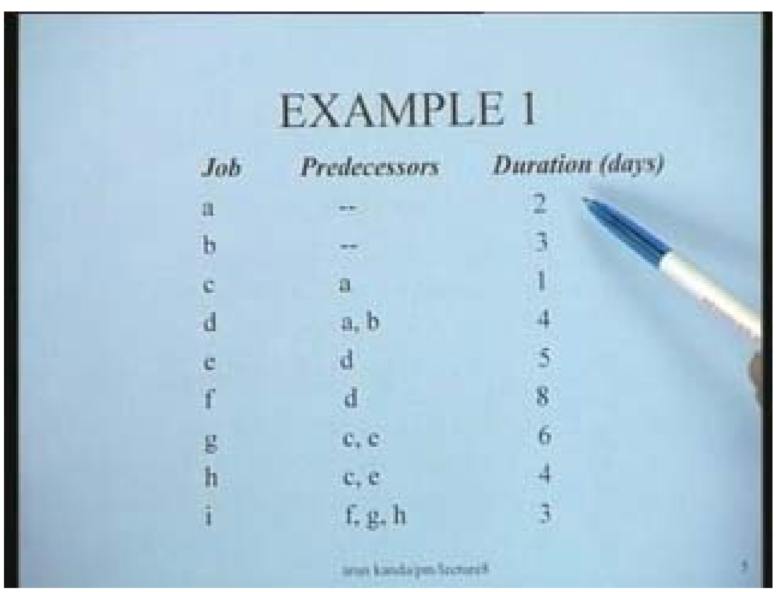
In such situations it's not always easy to estimate the time of the activity quite accurately and therefore you can assume various estimates of time and as far as the manner in which the estimates are obtained here is what we can say. In deterministic times for instance a single time estimate is used for each activity and this is taken from experts who have prior knowledge and experience of the activity. We are working only with the single time estimate for each activity whereas in probabilistic times three time estimates are used. The optimistic, most likely and pessimistic and for each of these activities these are generally obtained based on a consensus of the group of people who are working on these various jobs.

(Refer Slide Time: 5:28)



Let’s take an example and the predecessors for each of these jobs is identified as shown in this list and we are assuming that for each of these jobs we have estimates of the duration of the job in days and these estimates are given here in this particular list and the objective of the analysis or the basic scheduling computations is to establish a time schedule for various activities and to find out when this project can be accomplished at the earliest.

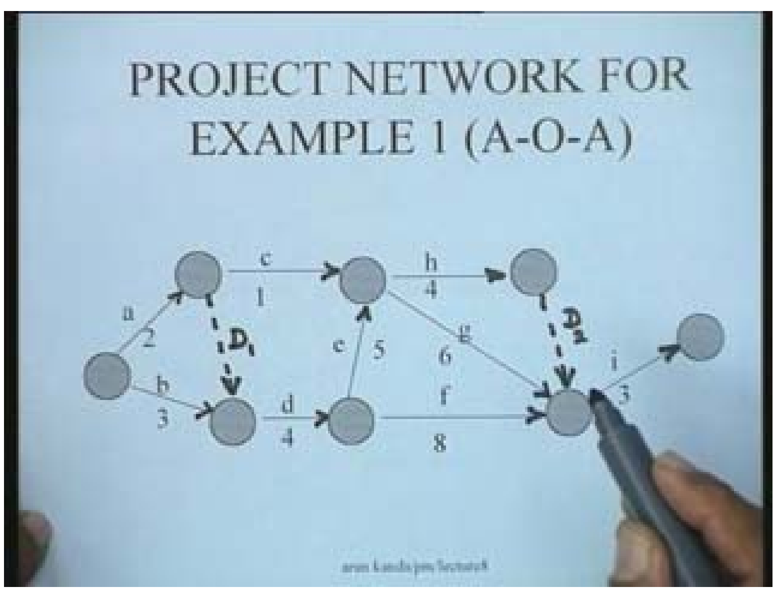
(Refer Slide Time: 6:17)



This is the broad intention of the whole analysis. The first step in doing this problem is to represent this information on jobs and predecessors in the form of an A-O-A network and this can be done very conveniently by means of a diagram. If you look at the list it would be something like this. In this particular diagram the activities are shown here. This is activity a, activity b. If you look at the precedence relationships of the jobs, a and b had no predecessors. They are starting from a source node. C has predecessor a as shown here but activity d has predecessors both a and b. We need to add a dummy activity from here to here showing that activity d now has predecessors both a and b as per the requirements. Then we talk about activity e. Activity e has predecessor d as shown here. Activity f also has predecessor d. Activity f is shown here. Then activities g and h they both have predecessors c and e. So c and e are the predecessors for both g and h as shown here and as far as activity i is concerned it has three predecessors namely f, g and h.

We can bring a dummy arc from here and show it like this and then f, g and h are all shown as predecessors for activity i. A comment on this particular network is in order; you would recall that when we were talking about various dummies. We can identify in this particular case that we have used 2 dummies. You may call them D1 and D2. Notice that D1 is a logical dummy and what is the nature of the dummy D2? It is not a logical dummy but it is a dummy which is necessitated by the fact that activities g and h are in parallel and therefore for uniqueness of activity representation I had to introduce this particular dummy D2.

(Refer Slide Time: 9:20)

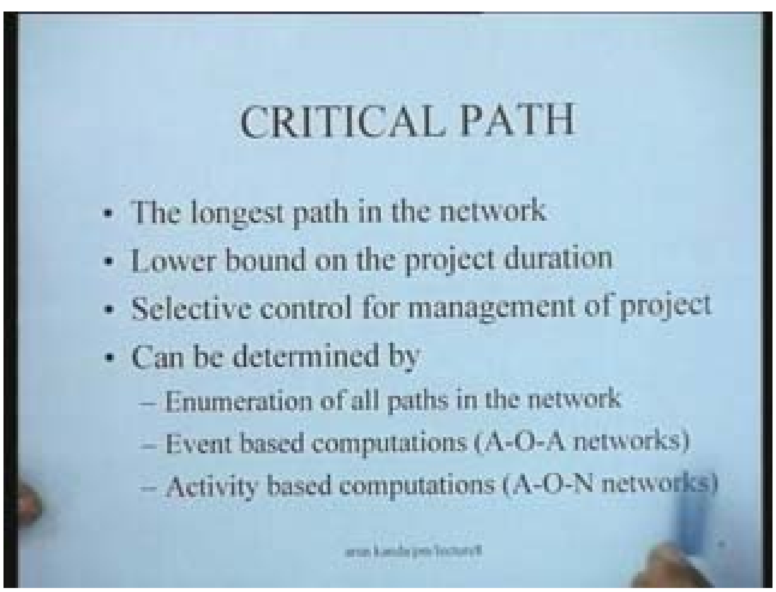


These are the 2 dummies which have been introduced in this particular network and we have indicated here the activity durations adjacent to the arcs. Notice that in the A-O-A representation it is the arcs which represent the activities. So the durations of the activities are shown adjacent to the arcs in this particular case here. Now in this particular example our objective is primarily to determine the critical path.

How do we determine the critical path and what is the critical path? What is the significance of the critical path for a project? A critical path is in fact the longest path in the network and it is the longest path in the network Therefore interpretation of the critical path is that it is lower bound on the project duration; that the project cannot be accomplished in a time smaller than the length of the critical path. A project may take longer but this is the minimum possible duration of the project. Another important thing about the critical path is that it is a means of providing selective control for management of the project. A critical path by the virtue of the fact that it is the path which determines the project duration identifies a subset of activities and the management by concentrating on those subset of activities can in fact monitor and control the entire project.

It is something like A class items in inventory control. When you talk about ABC analysis in materials, you talk about controlling the entire cost by concentrating on A class materials. Similarly here by concentrating on the critical activities you can monitor and control the project duration and to a very large extent the project cost as well and this critical path can be determined by a variety of ways. One of the simplest methods you can say or a naïve method for determining the critical path is by enumeration of all the paths in the network and this would give you not only the longest path but the next longest path and all the paths in the project network and there could be many applications in project network analysis where one might be interested in not only the critical path but the sub critical paths. For instance when one is doing project crashing, the project crashing simply means that the project is being crashed by spending more resources on activities. Activity times are being reduced. When activity times are reduced first the critical path has to be made shorter and when it is made shorter other paths become critical. You have to keep a track of the various other paths as well. Enumeration is something that can be done or event based computations on A-O-A networks is another way of dealing with determination of the critical path or we could have activity based computations as one tries to do in activity on node networks.

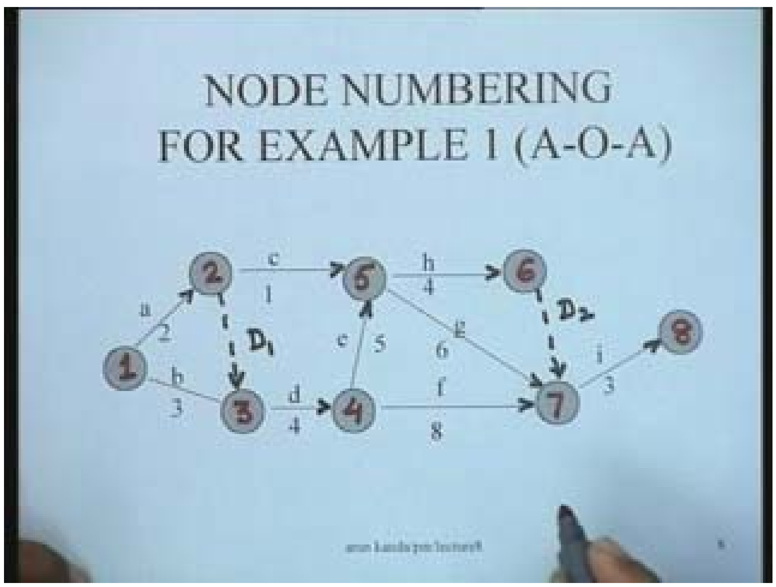
(Refer Slide Time: 13:05)



We shall try to see how this computation can be done. But a prerequisite for a situation of that kind is really that we have our network. Let us recall the network that we had drawn. We had drawn this network and let's first try to do a node numbering according to Fulkerson's rule. In this particular network we have these 2 dummies. We call them D_1 and D_2 and we had these various activities in the project network and what we would like to do is we would like to establish the node numbers. The nodes are not numbered. The numbering of nodes is in fact the first crucial step before proceeding with the algorithm. In this case there is only 1 source node.

Let us try to number the nodes of the network. The source node is numbered as node number 1. This would be clearly node number 2 and after this, this would be node number 3. Then we can number this node as node number 4 and then we can call this as node number 5 and after this we have to call this as node number 6. Then we have got to call this as node number 7 and finally this is node number 8.

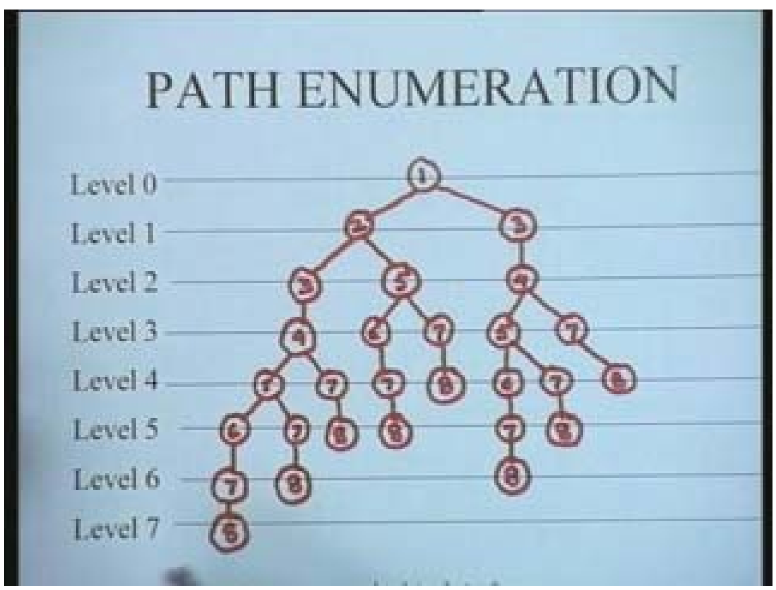
(Refer Slide Time: 14:36)



This is the node numbering scheme and we have been able to now develop a network in which all the nodes are properly numbered. For instance look at this particular network. It has a single source and a single sink and what we are interested in finding out is how do we find out the critical path in this particular project? I indicated to you a minute ago that determination of the critical path can be done by enumeration of all the nodes. Let us take this network and see if we could determine the critical path in this network by nodes. The path enumeration can be accomplished by using a tree like structure and what can be done is let us try for instance the starting node of this particular network is node number 1.

I begin with the starting node 1 here. From node number 1 in the network there is an arc which goes to node 2 or there is an arc which goes directly to node number 3. I can identify that from 1 I can go to either 2 or I can go to 3. When you look at the network from 2 you can go either to 3 or to 5. From 2, I could go either to 3 like this or I could go to 5 like this and if I look at the node number 3 from node number 3 I can go only to node number 4. This is the only possibility. Then I look at these individual nodes. From node number 3, I can go only to node number 4 and from node number 5 there are two possibilities. I can go to either 6 or 7 as far as the network is concerned and from node 4, I can go to either 5 or 7. I can go to either 5 or 7 in that particular way. Then from 4, I can go to either 5 or 7 and from 6. I can go only to node 7. That is only one path from 6 to 7 and from 7 I can go only to 8. In fact when you reach 8 it is the terminal node. It is the last node in the network. That identifies that this is the end of it. I cannot go any further from this particular node. You can look at node 5. From node 5 I can go to node 6 or I can go to node 7 and from node number 7 I can go to node number 8. As far as 8 is concerned it's over. These nodes will not be tapped further. But from 5 I can go to 6 and 7. I can go to 6 and I go to 7. From 7 I can go to 8. From 7 I can go to 8. From 6 I can go to 7 and from 7 I can go to 8. Then finally from 6 I can go to 7 and from 7 I can go to 8 and these I cannot go further. From 7 I can go to 8 and as far as this entire level is concerned it has been explored. When you come to level 6, now you find that from 7 you can go to node 8.

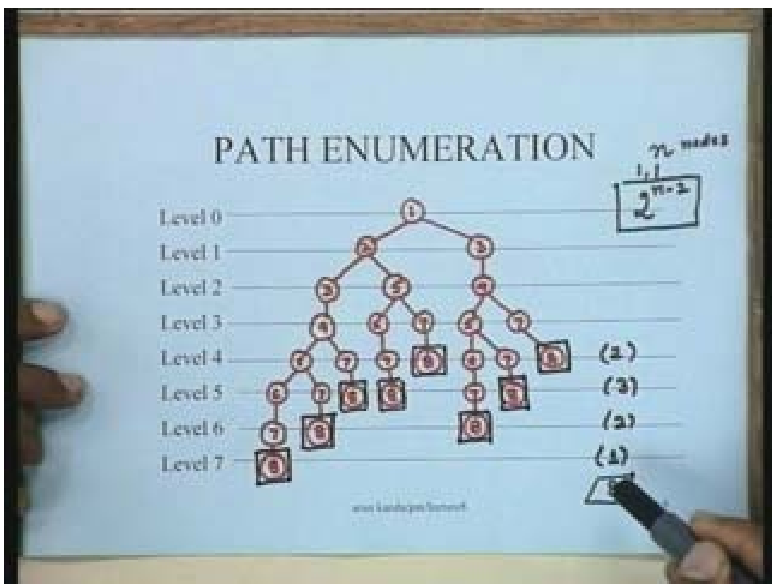
(Refer Slide Time: 19:08)



What I have accomplished is actually I have been able to enumerate all the nodes in the network and what you find is that when you come to node 8 this is a terminal node. This is a terminal node. This is a terminal node. This is a terminal node because 8 is the final sink in the network. This is a terminal node. This is a terminal node. This is a terminal node. This is a terminal node. This is a terminal node. What it really shows is that at level 4 in this entire enumeration exercise I have been able to discover 2 paths. This and this and these two paths are 1 2 5 7 8 and 1 3 4 7 8. I have been able to discover these 2 paths. At the level 5, actually level 5 means that each of the paths at level 5 would have 5 arcs in it. There are 3 paths at this particular level and there are 2 paths at this level and there is 1 path at this level. The total number of paths that has been discovered in this particular path enumeration exercise has been 2+3+2+1=8. There are actually for this example 8 paths from source to sink in the network and they have all been enumerated here.

This is a systematic tree enumeration algorithm which can be used for enumerating all the paths. We could also keep track of the individual path lengths and determine which out of these 8 paths is the longest, which is the shortest and so on and in this process we could determine the critical path. This is the process of determining the longest path in the network by path enumeration. The major difficulty with this approach is that it is time consuming and moreover the number of paths in a project network can be very large. For instance you have a network with n nodes with a single source and a single sink. The upper bound on the number of paths is 2 raised to the power n-2. This is the upper bound on the number of paths that can be there in a network which has n nodes with a single source and a single sink and this number can be very, very large. For this particular example the upper bound on the number of paths is 64 whereas there are actually 8 paths in this network.

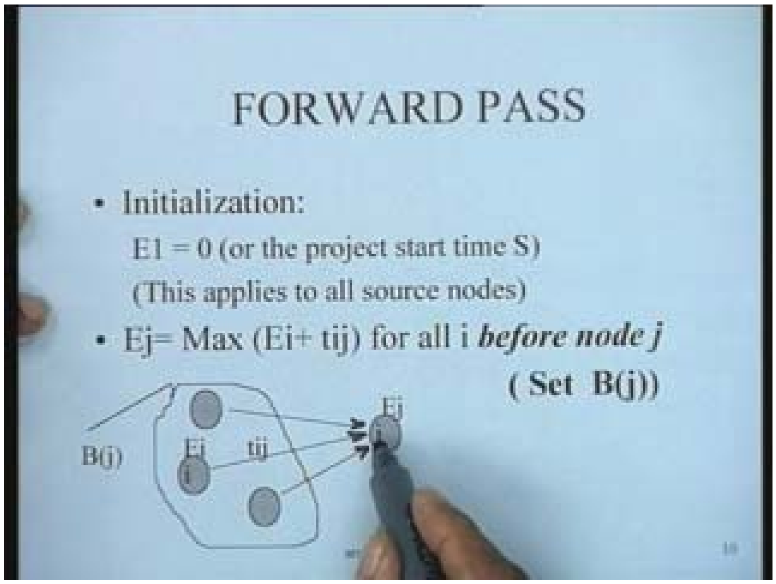
(Refer Slide Time: 22:02)



This method of path enumeration is really not a very practical method for determining the critical path. But nevertheless if one is interested in finding out all the paths in the network one could use this to determine all the paths and their various path lengths. It can be used from that point of view.

Looking at this particular problem what we are going to do is an event oriented algorithm for determining the critical path and this can be determined. Basically in any network we do what is known as a forward pass and the forward pass consists of essentially one initialization step. The initialization step simply says that the earliest occurrence time of the first node is equal to zero or it is equal to the project start time S. This is the initialization time of the project. This applies to all the source nodes. If there are many source nodes in a project network they would all have an earliest occurrence time of the initial event equal to zero and then the basic computation is being performed by this particular formula. We are actually interested in determining the earliest occurrence times of all the nodes. If this is node i, which has an earliest occurrence time of Ei and if this is node j which has the earliest occurrence time of Ej then essentially we have all these arcs which culminate in node j. These set of nodes are the immediate predecessor nodes of this particular node j.

(Refer Slide Time: 23:57)

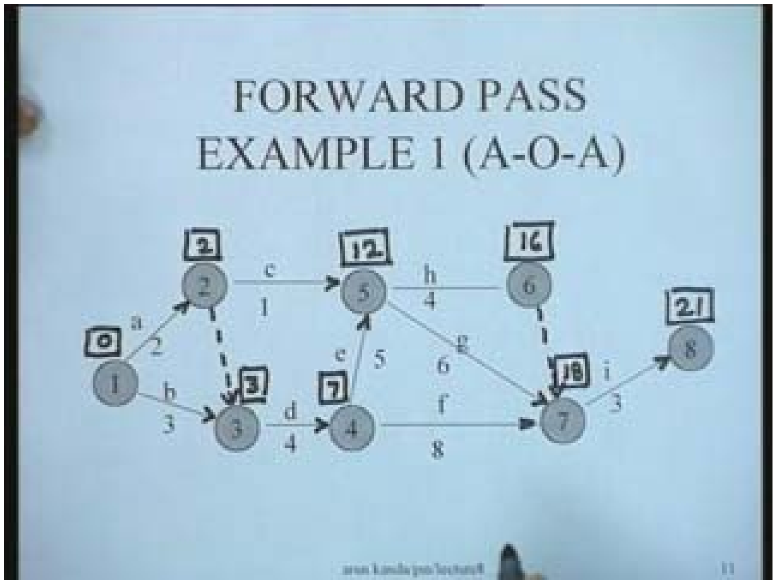


We call this set as the set B(j) or the set of nodes before j. What we basically do is we estimate Ej as the maximum of the earliest occurrence time of this node plus the duration of this activity and we compute the maximum over all these nodes and that would in fact define the value of Ej for this particular network.

Now let us try to apply this to the sample network which we are trying to investigate. We have this network here with us. We have the nodes numbered according to Fulkerson's rule and we would like to do the forward pass. This particular network has these activities, a dummy node here and then it has a dummy activity here and the various activities are as shown in this particular network. Let's do a forward pass and try to estimate the node time. There would be only 1 source node. We give a starting time of zero to this particular node; so earliest occurrence time of this node is zero. Then we would simply say 0+2 is 2. We give an earliest occurrence time to node 2 of 2. Then we would come to node 3. Node 3 has 2 predecessors. We have to take the maximum of 0+3 which is 3 and 2+0 because the dummy takes zero time. The larger of the two is 3. So this is 3. We find out the earliest occurrence time of node 3 in this particular manner.

Then we come to node 4. 3+4 is 7. Then you look at node 5. We have to take the maximum of 2+1 which is 3 and 7+5 which is 12. The maximum is 12. The earliest occurrence time of this particular node will be 12 and then you come to node 6. One advantage of the Fulkerson's numbering rule is that you can compute the earliest occurrence times in the sequence of the node numbers because this automatically determines a topological order in it. 12+4 would be 16 which would be determined here and then when you come to this node it has 3 predecessors. So 7+8 would be 15. 12+6 would be 18 and this 16+0 would be 16. So the largest is 18. This number would be 18 and 18+3 is 21. This is the completion of the forward pass.

(Refer Slide Time: 27:06)

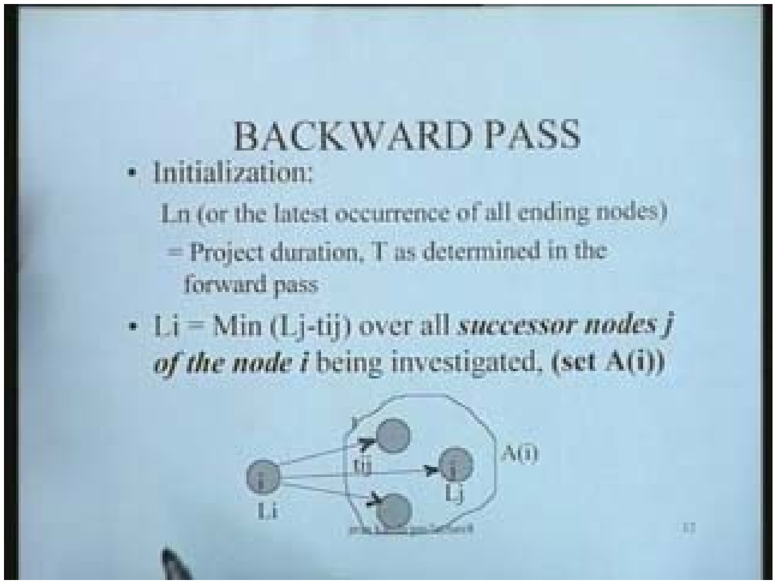


The forward pass indicates that the project duration is going to be 21 days. That's the minimum time in which the project can be accomplished.

Now let's see how we compute what is known as a backward pass. Let us first define the basic formulas for doing a backward pass. A backward pass also has an initialization step. The initialization step simply means that the latest occurrence time of the terminal node Ln is nothing but the project duration T as determined in the forward pass. In the example that we were doing the project duration was 21 days and once you have determined this, you have done this initialization the next step is to determine the latest occurrence times of all the nodes Li's which is nothing but minimum of Lj minus tij over all the successor nodes j of the node i being investigated. This is the set of nodes after i. If I am now investigating node i I look at all the successors.

Successors would mean that they would be directed like this and I take a particular node whose latest time I know; Lj minus tij. I would compute this for each of these successors and take the minimum of that and that would give me the Li value for this particular node.

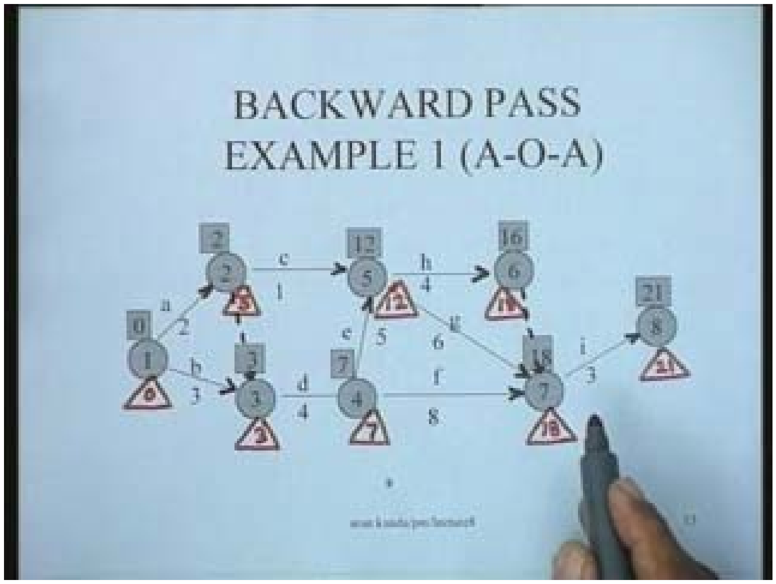
(Refer Slide Time: 28:42)



This in essence is what we mean by a backward pass. Now let's see how the backward pass is done. We have already done the forward pass for this particular network. In the forward pass we have seen that we have identified the numbers which are written in the top boxes of the particular case. In this particular example let us take this particular network. Recall that this particular network has a dummy here. It has another dummy here which goes in and the directions of the arrows are as they are shown here and what we would like to do is after having done the forward pass for which the results are shown in the boxes on top of each of the nodes we would like to do a backward pass and when you do a backward pass you start from the terminal node and the initialization step specifies that since the project duration is 21 we would initialize the value of the latest occurrence of this particular node to equal to a value of 21 and I put it in a triangle just to distinguish it from the early occurrence times. Then you can compute this value. 21-3 is going to be 18. I can put this in the triangle here.

Then we come to node 6. This being a dummy activity is zero. So 18-0 is 18. This particular value is just 18 and I put it in the box. Look at node 5. Node 5 has 2 successors. In the backward pass we are concerned with the successor activities. It has these 2 successors. So I have to look at 18 and 18 and I have to say 18-4 is 14 and 18-6 is 12. The minimum of the two is 12. The latest occurrence time of this particular node 5 is 12. Look at node 4. Node 4 would have 12-5 is 7 and 18-8 is 10. The minimum of the two is 7. So I have 7 listed here. Let's now take node 3. 7-4 is 3 and as far as node 2 is concerned it has 2 successors. So I have to do 12-1 which is 11 and 3-0 which is 3. This particular node value is going to be 3 and ultimately 3-2 which is 1 and 3-3 which is 0. So this is zero. The numbers and the triangles at the bottom of each node establish the latest occurrence time for each of the nodes.

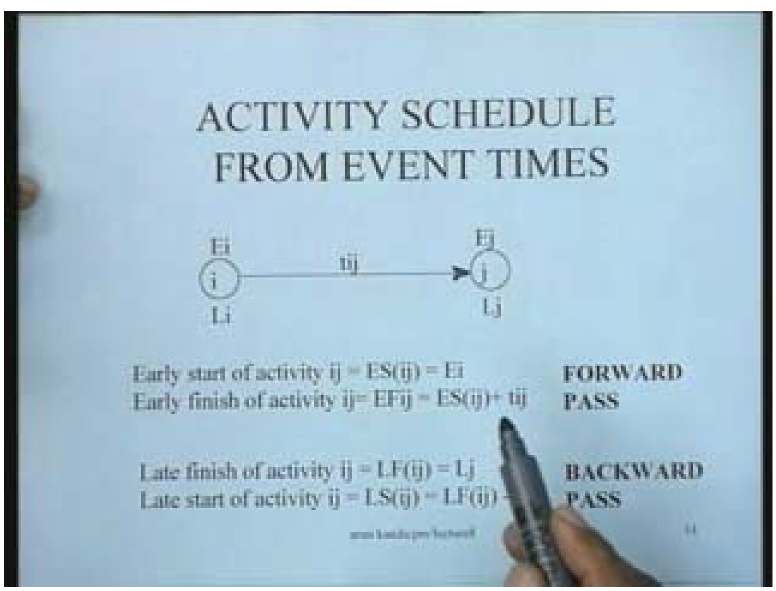
(Refer Slide Time: 31:40)



Their computation has been illustrated for this particular example.

The next step is we are interested in determining an activity schedule from the event times. The logic that we would have to use is simply that for each activity which goes from i to j in this particular manner we have been able to determine earliest occurrence time, the latest occurrence time, the earliest occurrence time and the latest occurrence time both the predecessor and the successor node for each activity. 3216

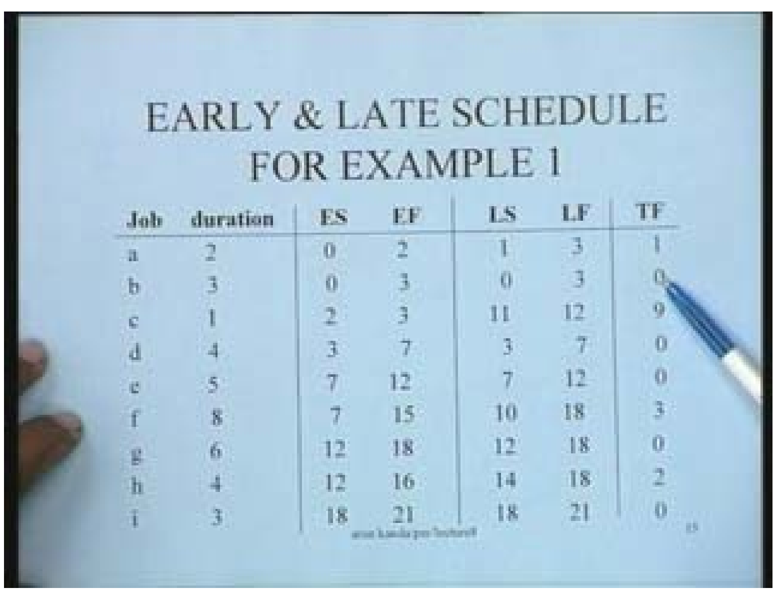
(Refer Slide Time: 32:16)



We use this information and try to determine the early start of the activity ij. Early start of the activity ij is nothing but the earliest occurrence time of its predecessor node. It's just Ei and once we have determined the early start of the activity, the early finish of the activity is is early start plus the duration of the activity and remember that the early start and early finish of the activities both of them have come from the forward pass. Similarly the late finish of the activity is determined by the latest occurrence time of the terminal node, this one. The latest finish of the activity ij is nothing but Lj, this particular value and therefore the latest start of the activity ij is determined by subtracting from the latest finish the duration of the activity and so you get the latest start of the activity. This information is actually available from the information from the backward pass. The forward pass and the backward pass are determined to give us the information pertaining to the early start and early finish, the late start and the late finish for the activity.

If we utilize the information that we have just obtained for the forward pass and backward pass for each of the activities a, b, c, d, e, f, g, h and i whose respective durations we have with us, we can identify the early start of the activity and from the early start of the activity we can add the early finish. We can add this duration to obtain the early finish of the activity and similarly the latest finish of the activity is obtained from the node occurrence time of the terminal node, late node occurrence time of the terminal node and this is the late finish of each of these activities and by subtracting these durations from here we get the late start time of these activities and we can define a quantity called the total float which is nothing but the difference between the late start and the early start or the late finish minus the early finish of the activity and by this process we have been able to determine the floats, the total floats for all the activities and the activities which have zero total float.

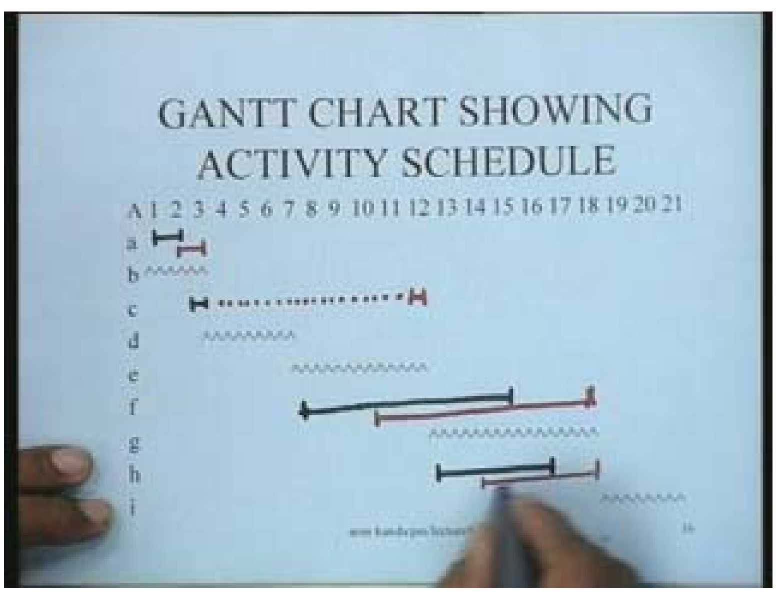
(Refer Slide Time: 35:05)



For instance activity b or activity d or activity e or activity g or activity i which have zero total floats in this manner. These are the activities which are critical activities or activities that lie on the critical path. This information which is the basic schedule determined by this exercise can also be easily obtained on the basis of what we call a Gantt chart. A Gantt chart is a convenient representation. In fact many computer programs automatically give you a Gantt chart representation of the project of the schedule that you have. In this particular case what we find is that these activities b, d, e, g and i are critical activities. So they have no floats. Their values are fixed and they must occupy the time slots as shown and you will find that the durations of the critical activities add up directly to the project duration of 21 days. If you look at activity a, activity a for instance has a duration of 2 days and the early start of this activity if it is done in the first day or in the second day it would be something like this. There is a possibility of delaying this activity to the first and the third day. It can actually be done here or here. Activity a can earliest be done in this particular time period or it can be done in this particular time period. Similarly let us look at activity c. Activity c at the earliest can be done in the third period. So it has duration of 1 day. It can be done earliest here or it can be delayed and it can be delayed by 9 days so that it is accomplished in the twelfth day. Ultimately it can be done in this particular time period and in fact it has a lot of flexibility. It can be done either here or here depending upon where it is.

If you now look at activity f, activity f can be done. Activity f has duration of 8 days and it can be started in the eighth day and it can continue. The earliest start is the seventh day. So seventh to the fifteenth day it will go on in this particular mode here, something like this or it could be actually rescheduled so that it is completed 3 days later in period 18. So it is started here in period 10, something like this. This particular activity can be slided from here to here and it would not really affect the total project duration if this was done in this particular manner. If we look at the other activity 8, h can be done from period 12 to 16. It can go on from period 12 to 16. That's the early start schedule for this particular activity or the late start schedule for this particular activity can be going on up to period eighteen and from here something like this.

(Refer Slide Time: 39:13)

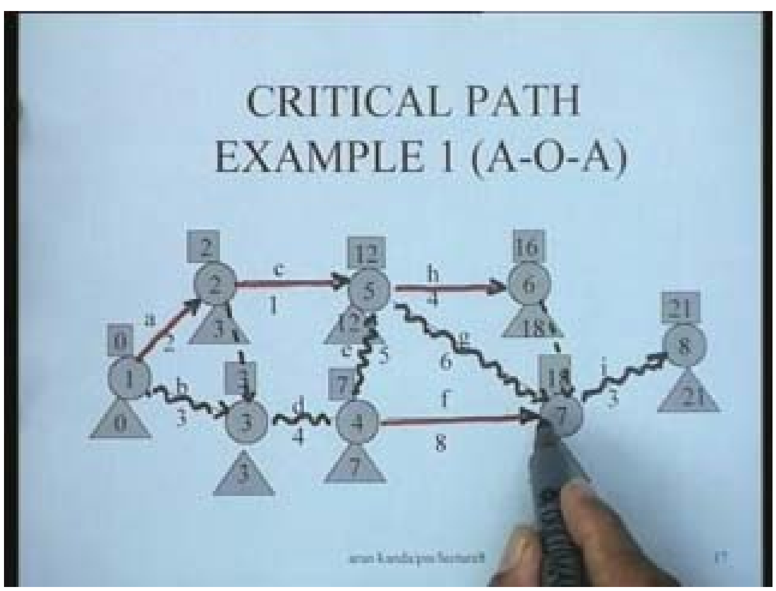


The point really is that a representation like this of all the activity schedules which show the early start and the late start possibilities for all the activities can be very conveniently shown in the form of a Gantt chart and this would show when which particular activity is scheduled to be done.

Let us now make some comments on the critical path. In this particular case the computations that we have performed both for the forward pass and the backward pass are as shown in this particular network and recall that there are dummies here and there is another dummy going from here to this particular node and this is the project network which we have been analyzing so far. If you look at this project network how would you identify the critical path? Remember we just did the analysis and the critical path would be something like this. Activity b was critical and thereafter d was critical. Then e was critical. Activity g was critical and finally i was critical. The point that I just wanted to stress here was that the critical path has been identified after the computation of the total floats and those activities which have zero total float have been identified in this list and they constitute a path b, d, e, g and i in the project network which is the critical path.

The interesting thing to note here is that it is not always necessary that an activity joining two critical nodes will be critical. For instance take this very network and see that node number 4 is a critical node. Node number 7 is also a critical node but this particular activity f is not a critical activity. Yet it is an activity joining 2 critical nodes and therefore the correct procedure for identifying the critical activities is first computation of the total floats and then identifying those activities which have zero total float. So this is not a critical activity. This is also not a critical activity. This is also not a critical activity. This is also not a critical activity and as far as the real activities these 4 activities are not critical and yet this is an activity joining 2 critical nodes.

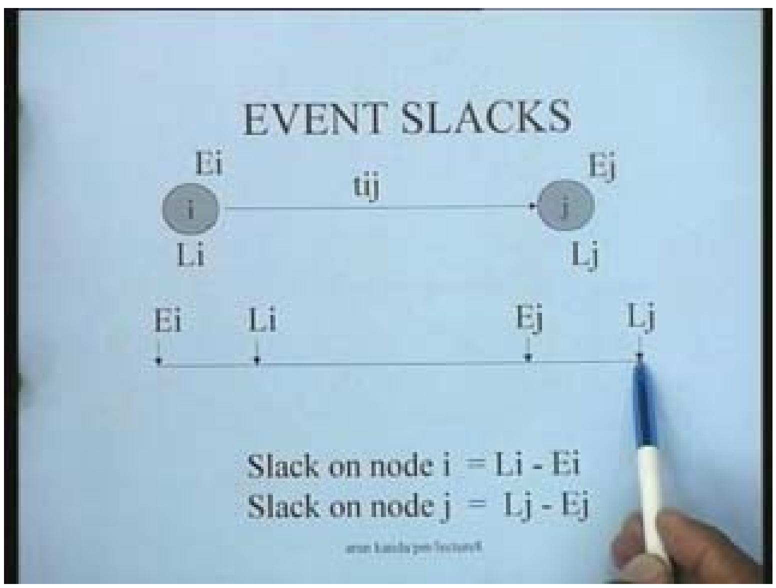
(Refer Slide Time: 42:12)



One has to be a little cautious in computing or in identifying the critical path because simply joining 2 critical nodes without verifying that this particular duration exactly matches this. This is not critical because 18 to 7. This is 11 units of time and this activity takes only 8 units of time. So it is not critical. It has a float of 3 units. This is how we can identify the critical nodes.

Let us talk about the important concept of event slacks and total floats for various activities. Let's first talk about event slacks. If we have an activity going from i to j we have computed through a forward pass the earliest occurrence times and through a backward pass the latest occurrence times for all the nodes. The earliest occurrence time would be earlier and the latest occurrence time would be later and if it's a critical node these would coincide and the earliest occurrence time for this node is here and the latest occurrence time for this node is here.

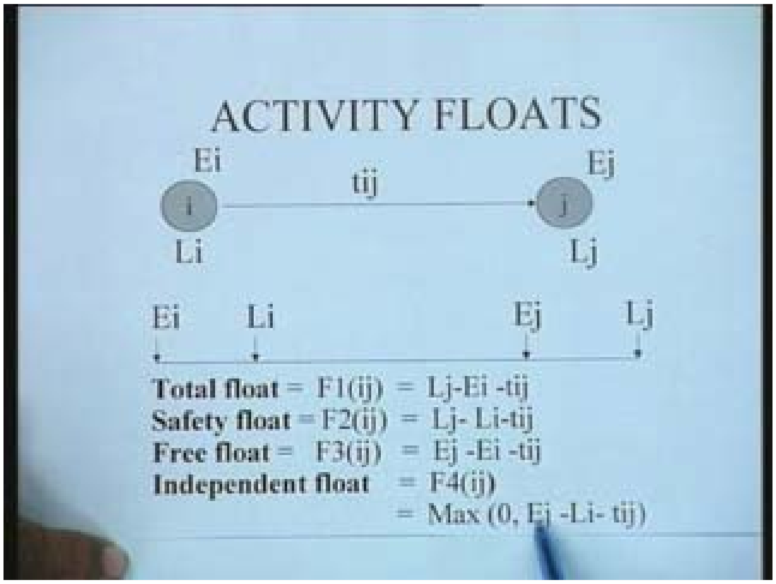
(Refer Slide Time: 43:20)



This is how they would show on a time scale. We can define this slack on a node. This particular node which is a milestone has a slack which is determined by the latest occurrence time minus the earliest occurrence time. So Li minus Ei is the slack on node i and slack on node j is Lj minus Ej. This is the definition of the node slacks. Then when we talk about the activities we can talk about activity floats. For instance this is an activity (i j). It's one activity; Ei Li Ej Lj. For each activity we can define 4 kinds of float. Let's talk about the total float first. The total float of the activity is defined as Lj minus Ei. It means that you are getting the maximum time duration for performing the activities Lj minus Ei minus tij and this is the maximum possible flexibility that can be given as far as activity (i j) is concerned and this quantity so computed is defined as the total float. Similarly the safety float is nothing but Lj minus Li rather than Ei; Lj minus Li minus tij because what the safety float assumes is that the predecessor activities have utilized their float and you are reaching this particular point at the latest time. This is defined as Lj minus Li minus tij

The third kind of float which is the free float for the activity is defined as Ej minus Ei minus tij. What it assumes is that you are trying to depart from this here after completing this activity as early as possible. You are not eating into the floats of the succeeding activities and moreover you have reached here as early as possible. That is the physical significance and the independent float is the worst case situation where you are assuming that you are reaching here late and you want to depart here early. The time available is only Ej minus Li minus tij for doing the activity.

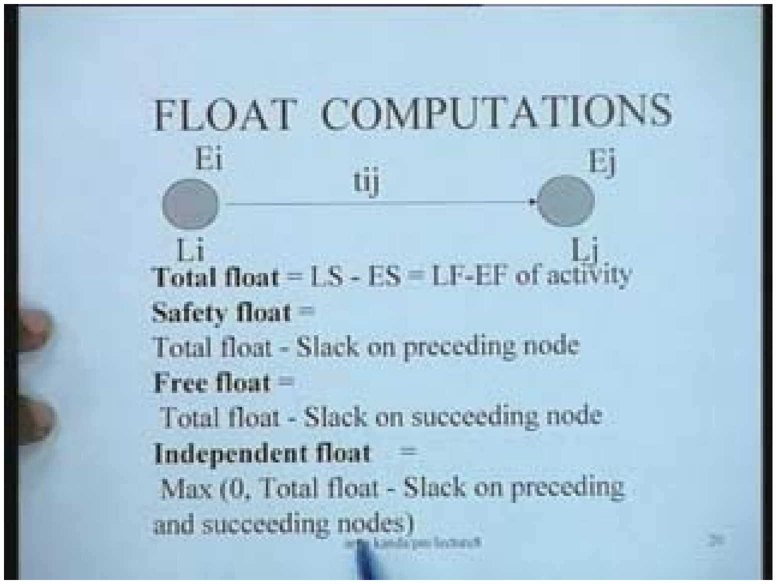
(Refer Slide Time: 45:37)



This time might not be adequate to perform the activity that is why this maximum of zero and Ej minus Li minus tij and this is a highly individualized measure of the activity float because it's assuming that preceding activities are done as late as possible and the succeeding activities are still done as early as possible and you can still manage to do the activity. That is the significance of the independent float.

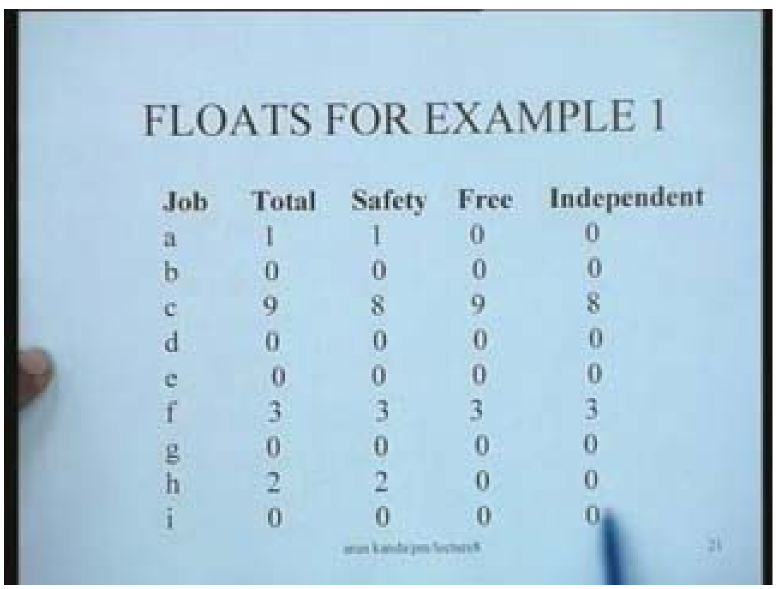
A convenient form for computation of these floats is that the total float is the way we calculated this; later start minus early start or latest finish minus early finish of the activity. For computing the safety flow what we can do is simply calculate the total float and subtract from it the slack on the preceding node that is Li minus Ei. Total float minus slack on preceding node will give me the safety float. The free float is obtained as the total float minus the slack on the succeeding node that is Lj minus Ej and the independent float is nothing but the maximum of zero and total float minus slack on preceding and succeeding nodes.

(Refer Slide Time: 47:00)



We try to determine the total float first and by identifying the node slacks we can determine all the 4 kinds of floats for that particular activity. For instance if you take the example that we have done for which we have already computed everything the total floats for this particular example works out for job a as 1. Safety float is 1. There is zero free float and zero independent float for the activity. Similarly for activity b. In fact for activities which are critical all the four floats are zero. b is a critical activity so is d; so is e, so is g and so is i.

(Refer Slide Time: 47:53)

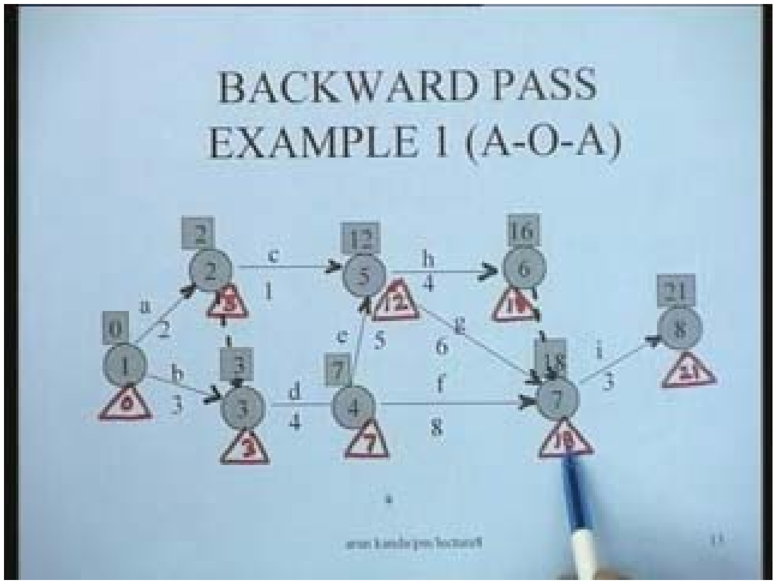


FLOATS FOR EXAMPLE 1

Job	Total	Safety	Free	Independent
a	1	1	0	0
b	0	0	0	0
c	9	8	9	8
d	0	0	0	0
e	0	0	0	0
f	3	3	3	3
g	0	0	0	0
h	2	2	0	0
i	0	0	0	0

Let's take for instance job f. If you see the computations of job f, we can determine for job f the total float. Let's refer to the calculations that we had done earlier. These are the computations we had done and look at this particular activity f. This particular activity f once we have identified the total float for this which is 3 the slack on the preceding node is zero. The slack on the succeeding node is also zero.

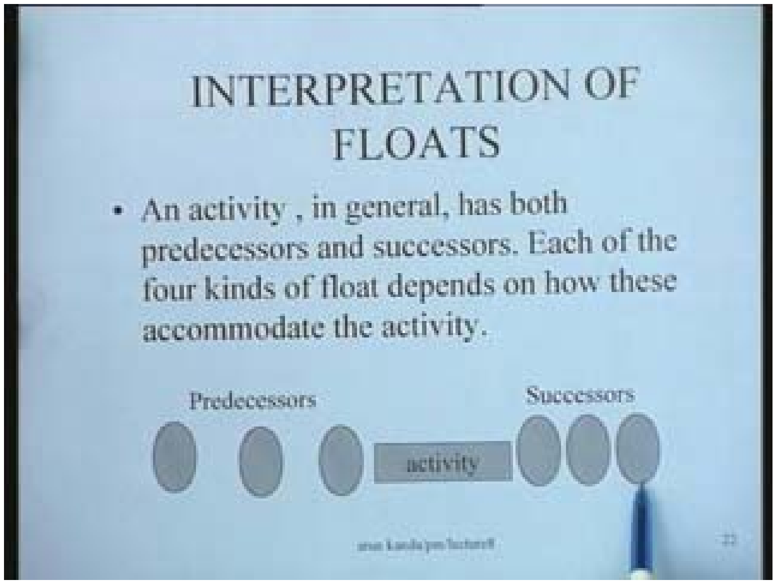
(Refer Slide Time: 48:44)



In this particular case all the 4 floats are going to be equal because slack on preceding node and slack on succeeding node is zero and so that is why we had for activity f all the four nodes ……. Take for instance the other activity h, this activity. For this particular activity the slack on the preceding node is zero. The slack on the succeeding node is 2. Since the slack on the succeeding node is 2 for this particular activity the total float is 2, slack on the preceding node is zero. So the safety float is 2 and the free float is zero because 2-2 is zero and the independent float is also zero. This is how we can actually compute all the floats of the example.

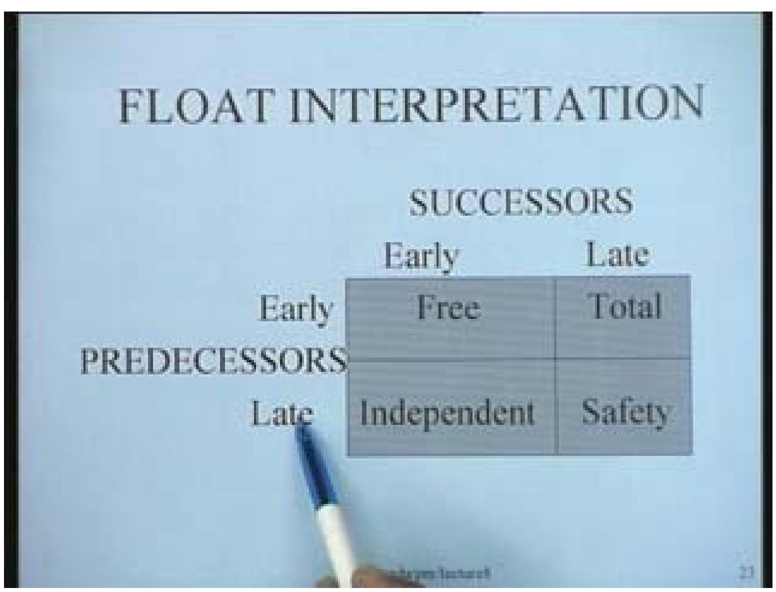
Let us now talk about some of the interpretations of these floats. The important thing to notice is activity. What is an activity? An activity in general has both predecessors and successors. So it is like a queue of people; some before it some after it and the activity is to be done in between and we constrain the duration of the activity which is the critical path.

(Refer Slide Time: 49:59)



An activity in general has both predecessors and successors and each of the 4 kinds of floats depend upon how these accommodate the activity. It depends upon whether these are shoved this side or these are shoved to the maximum possible extent. If these people are shoved to the maximum possible left hand side and these people are shoved to the maximum possible right hand side you would have the maximum amount of …….. for the activity and that is what governs the total flow and if these activities are not shifted to the maximum possible extent this diagram summarizes the various kinds of floats. The predecessors for the activity may be done either early or late.

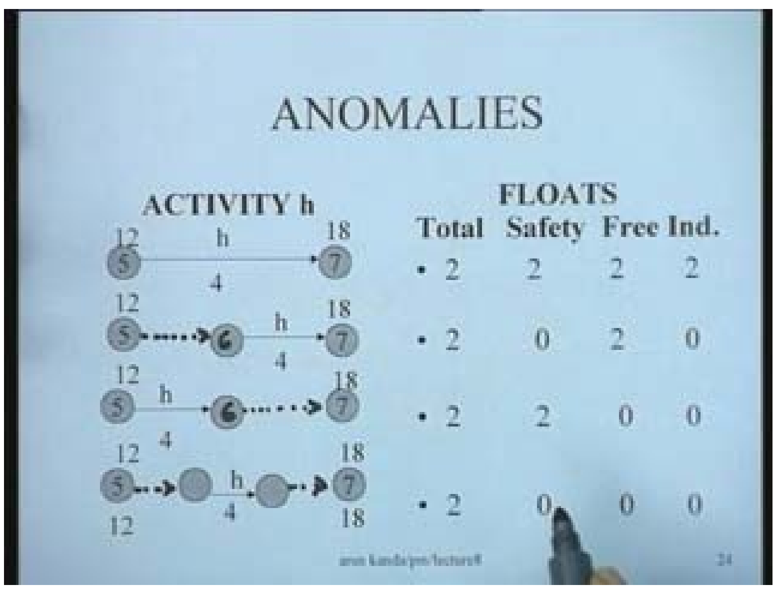
(Refer Slide Time: 50:43)



Similarly the successors for the activity may be done either early or late. Depending upon the possibility you have 4 types of floats. If the successors are done as late as possible and the predecessors are done as early as possible giving you the maximum, you have the total flow. Similarly if the successors are done as early as possible and the predecessors are done as early as possible we have free flow. If we have this possibility of the predecessors done as late as possible and the successors done as early as possible leaving you very little time that is the independent float and similarly the safety float is the case when you come to this particular box. This is the largest float. This is the minimum float and on the diagonal this way you don't know whether this is larger or this is larger. This depends upon the relative position of the network.

I would now like to talk about some anomalies. Look at the example that we did and let's talk about activity h which is one of the activities. We had activity h going from node 5 to 6, something like this and this was followed by a dummy activity which was going into node 7. Depending upon the placement of the dummy if h was a single activity without a dummy then these would be the node times. All the floats would be equal. If h was preceded by a dummy you find that the total float is 2, the safety is zero, the free is 2 and the independent is zero. On the other hand if this h is succeeded by an activity in this manner then it will be 2, 2, 0, 0 and if h is preceded and succeeded by a dummy in this particular fashion then you would have in fact a situation where it would have only total float and it would not have any of the other kinds of floats.

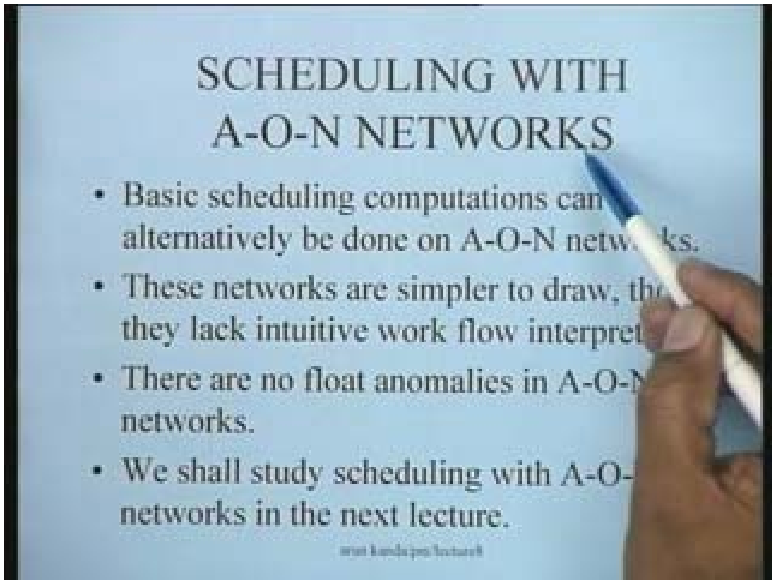
(Refer Slide Time: 52:59)



This is an anomaly because it's only 1 activity and the dummy is really placed there because it doesn't really matter whether you place here or whether you place the dummy here. But you see from this example that it does or does not? matter where you place the dummy and the interpretation could be that this dummy is equivalent to an activity which is actually constraining this activity because this could be like an approval activity which takes zero time, approval before you do the activity. This could be like a final examination or a quality control check before the activity is done which takes zero time. When this becomes a predecessor since this particular activity has an interpretation of whether this is how this is slid between the various activities it could have different kinds of interpretation. One really has to be careful while talking about some of these things.

One possibility is one can do scheduling with A-O-N networks.

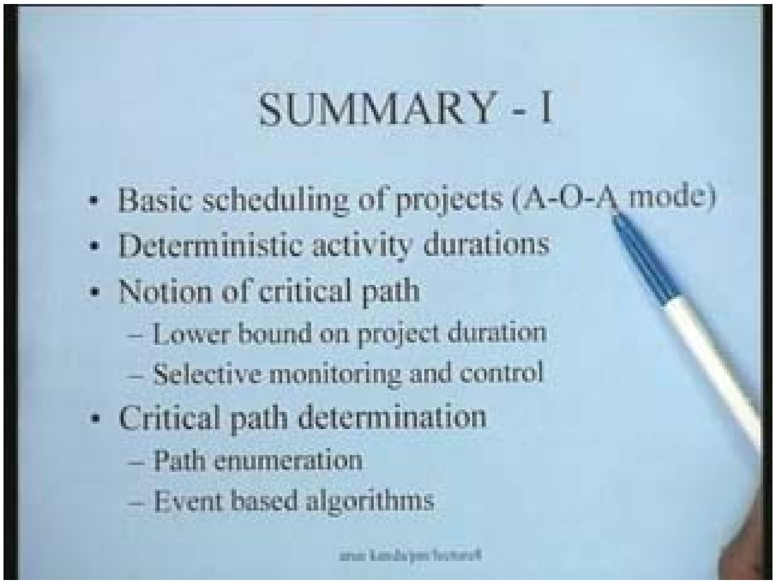
(Refer Slide Time: 54:00)



Basic scheduling computations can be done on the A-O-N networks. These networks are simpler to draw though they lack intuitive work flow interpretation and there are no float anomalies in A-O-N networks and we shall study scheduling with A-O-N networks in the next lecture.

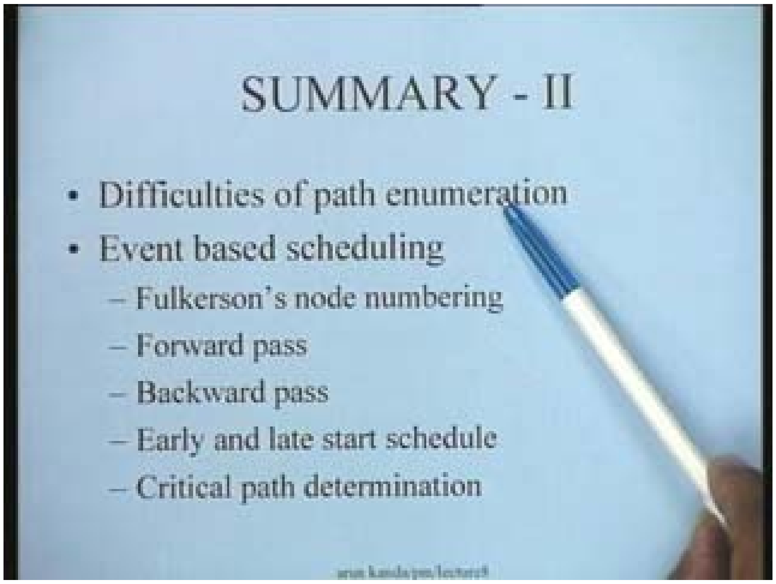
Finally let's try to look at what we did in this particular lecture. We looked at the problem of basic scheduling of projects in the A-O-A mode.

(Refer Slide Time: 54:31)



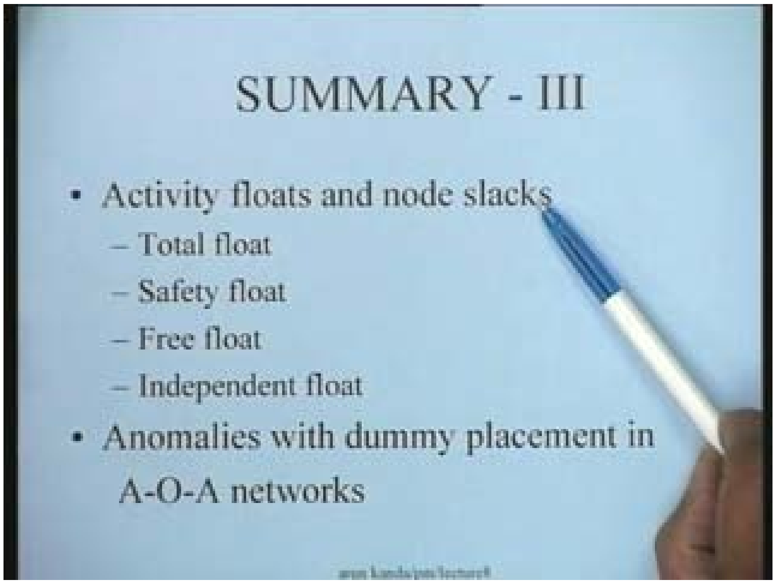
We dealt with deterministic activity durations. We talked of the notion of the critical path which is actually the lower bound on the project duration and which is also a device for selective monitoring and control. We looked at the process of determination of the critical path by using both a path enumeration algorithm and an event based algorithms which we illustrated by determining the earliest and latest occurrence times of events. We saw that there were difficulties in the path enumeration algorithm primarily because there could be a large number of paths and enumerating all the paths is difficult.

(Refer Slide Time: 55:12)



This particular method of determining the schedules was much more efficient. We had these basic steps; Fulkerson's node numbering, doing a forward pass, a backward pass, early and late start schedule determination and then determination of the critical paths and then we talked about some of the concepts of activity floats and node slacks.

(Refer Slide Time: 55:38)



We talked about the total float, the safety float, the free float and the independent float and we finally found that there were certain anomalies with dummy placements in A-O-A networks and these are to some extent removed when one talks about A-O-N networks which we will take up next time. Thank you!