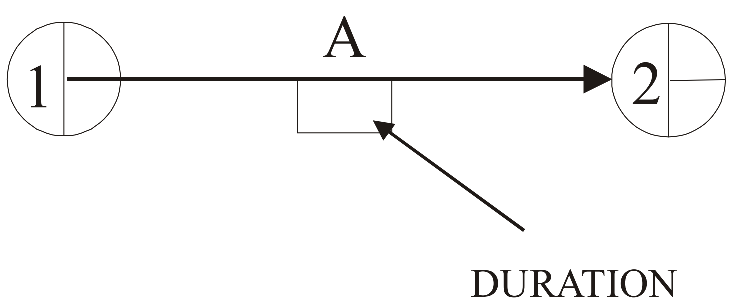
Critical Path Analysis

Key concepts

A network is a weighted diagraph. That means a diagram showing connections between vertices. The connections are called edges and a number is assigned to each edge, which is its "weight" or in this case "duration".

Critical path analysis commences with the description of a business process that involves a sequence of activities to each of which there is a duration. You are told for each activity what its preceding activities are and its duration. From this information you are expected to construct a network. In the process of constructing the network the vertices are introduced and numbered. The vertices are placed into circles with spaces to contain extra information; the duration of each activity is placed into a rectangular box attached to the edge representing the activity. For example

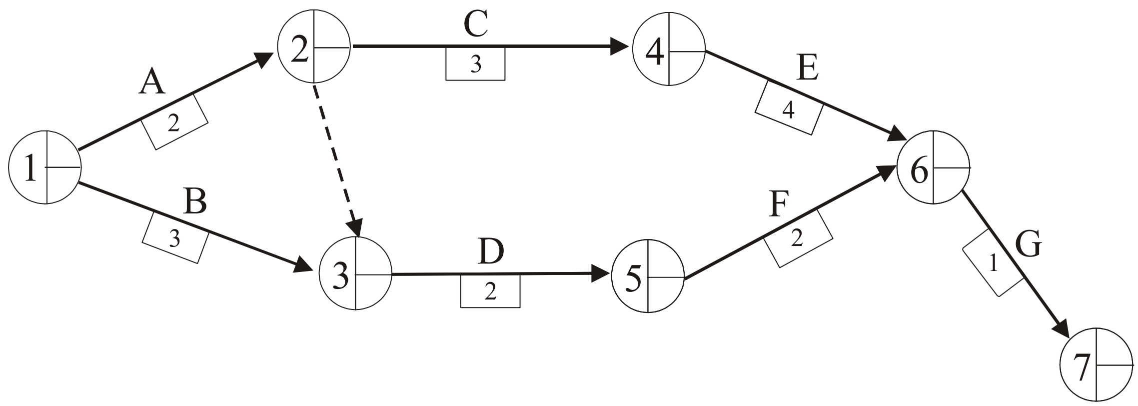


The two right hand boxes will contain additional information to be explained later.

In order to avoid ambiguity in drawing a network "dummy" activities may have to be introduced. For example, when drawing this network an ambiguity occurs.

ACTIVITY	*PRECEDING ACTIVITY*	*DURATION*
A	–	2
B	–	3
C	A	3
D	A, B	2
E	C	4
F	D	2
G	E, F	1

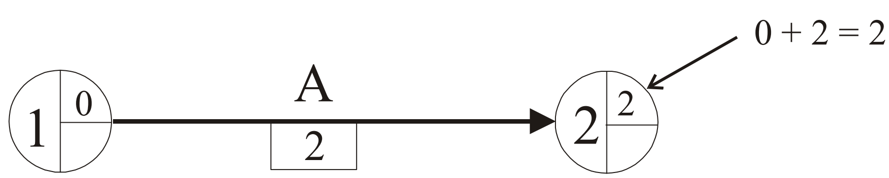
Activity *C* is preceded by just *A* and activity *D* is preceded by *A* and *B*. It is not possible to represent this information by a single node at the beginning of both activities. Consequently, a dummy activity must be introduced.



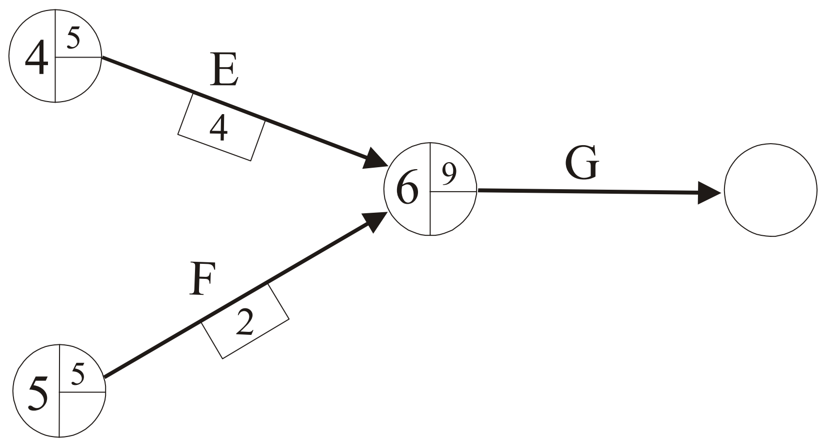
The dummy activity is represented by a dashed edge. The direction of the arrow tells us that activity *D* is preceded by activities *A* and *B*, whereas activity *C* is only preceded by activity *A*. The dummy activity has zero duration.

Forward Pass

The earliest start time of an activity is the earliest time at which an activity can start! Clearly, an activity cannot start unless all the preceding activities have been completed. Thus the earliest start time of an activity is the latest finishing time of the vertex that it starts from. This is worked out in the forward pass through the network. The earliest start time (EST) is recorded in the top right hand segment of each vertex. Starting with the first vertex the duration of the activity is added to the EST of the vertex from which it starts, and the result is placed in the top right hand segment of the vertex which completes the activity.



When two activities terminate in the same vertex the largest of the two possible numbers must be recorded, because the next activity cannot start until all the preceding activities have been completed.



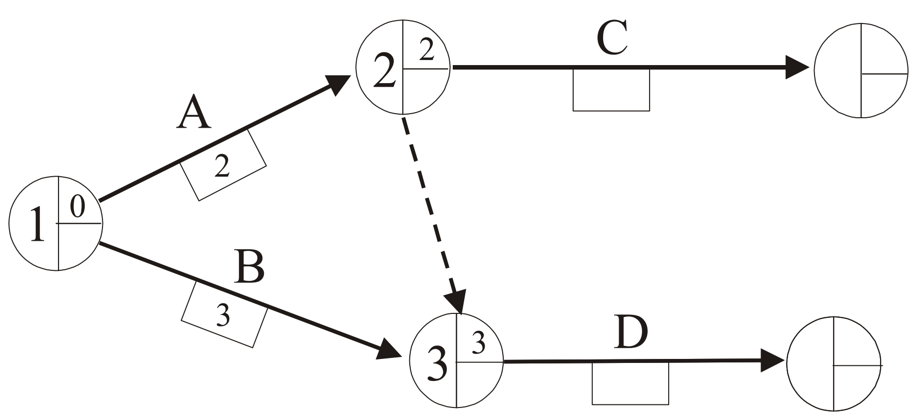
For E $\qquad 5+4=9$

For F $\qquad 5+2=7$

But G cannot commence until both E and F are finished;
therefore, the EST of G is 9

At the same time that we work out the EST of each activity we can also work out the earliest finishing time of each preceding activity. This is the earliest start time of the preceding activity + its duration. In the above example the EFT of E is 9, which is the same as the EST of G, whereas the EFT of F is 7. But note that these EFT values are not directly recorded on the network.

We deal similarly with vertices where there are dummy activities



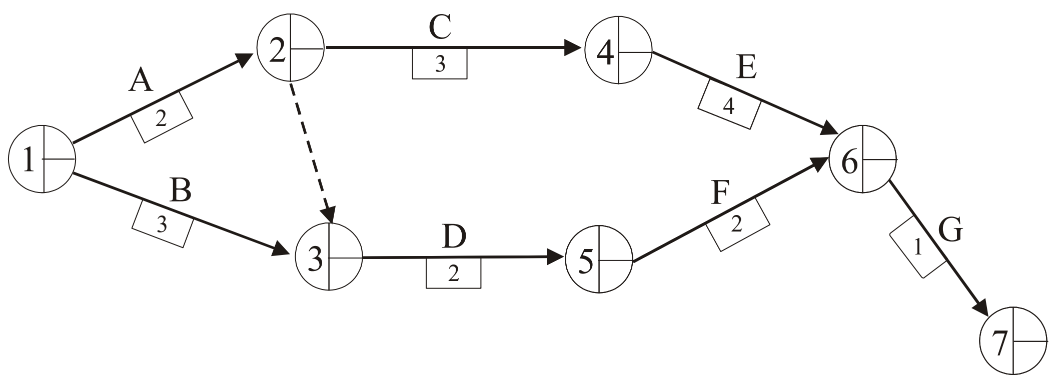
Activity D is preceded by both A and B . The EFT of A is 2; The EFT of B is 3; since D cannot start until both A and B are completed the EST of D is 3.

In the forward pass we compute the earliest start time and the earliest finishing times of activities according to the rules.

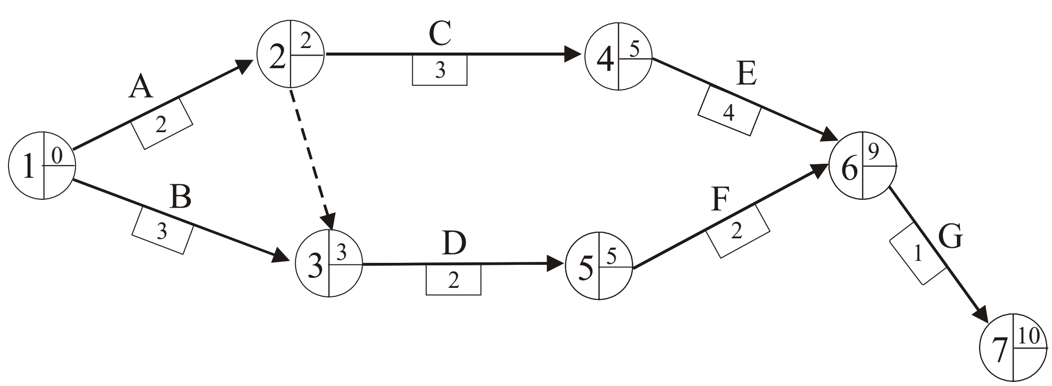
EFT = EST + Duration

EST = Maximum of EFTs of preceding activities.

A table is made to record the information that is being derived from the network.



Activity	Duration	Preceding Activity	EST	LFT	LST	EFT	TF	FF
A	2	-						
B	3	-						
C	3	A						
D	2	A,B						
E	4	C						
F	2	D						
G	1	E,F						

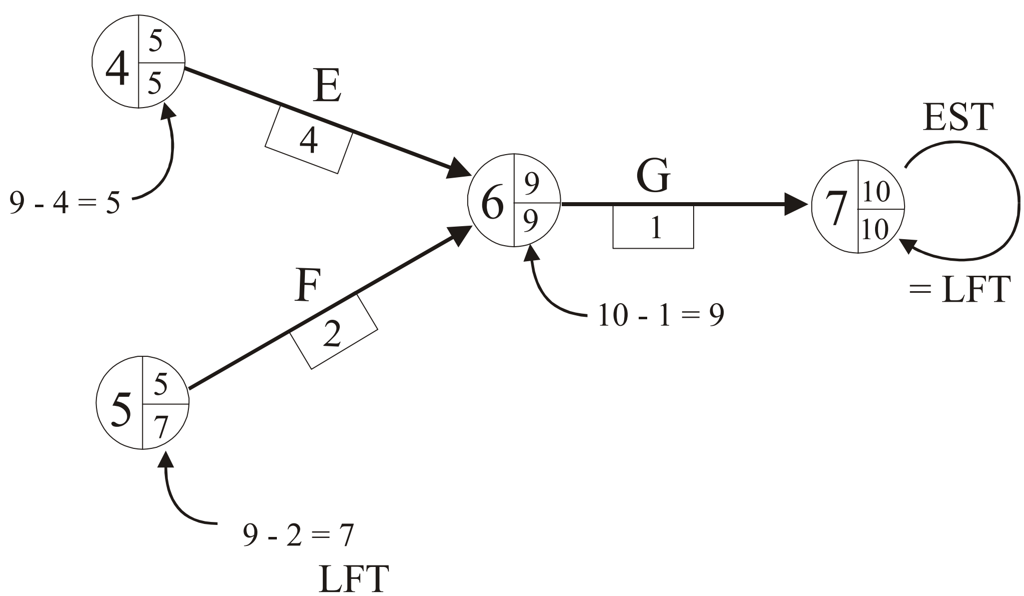


Activity	Duration	Preceding Activity	EST	LFT	LST	EFT	TF	FF
A	2	-	0			2		
B	3	-	0			3		
C	3	A	2			5		
D	2	A,B	3			5		
E	4	C	5			9		
F	2	D	5			7		
G	1	E,F	9			10		

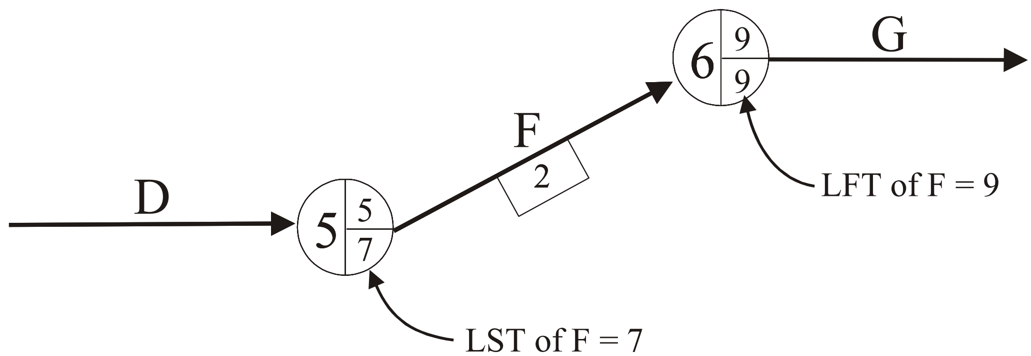
<u>Backward pass</u>

In the backward pass we compute the latest finishing time and the latest start time of each activity.

The latest finishing time (LFT) is the latest time at which the activity can finish without delaying the whole process. Thus at the final vertex the LFT is equal to the EST and from there we work backwards by subtracting the duration of each activity successively.

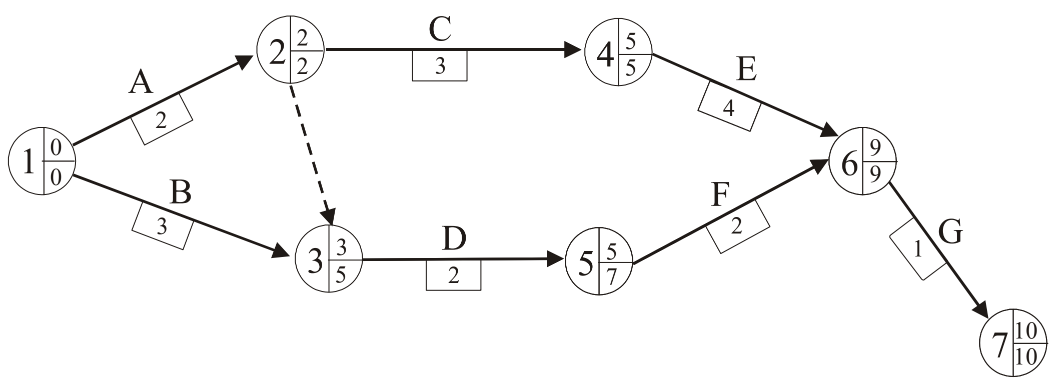


The latest start time is the latest time at which each activity can commence without delaying the whole process. In the above example the LFT of F is 9 and the LST of *F* is 7.



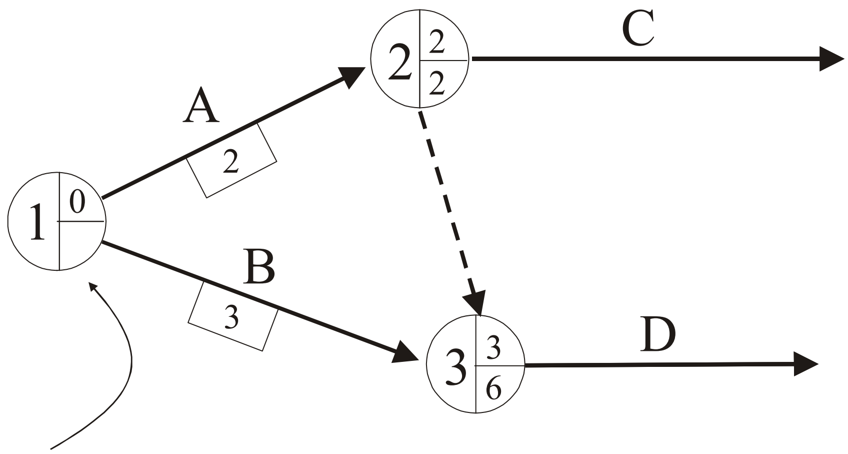
The LST of *F* is the LFT of *D*

The LFT of *F* is the LST of *G*



Activity	Duration	Preceding Activity	EST	LFT	LST	EFT	TF	FF
A	2	-	0	2	0	2		
B	3	-	0	5	2	3		
C	3	A	2	5	2	5		
D	2	A,B	3	7	5	5		
E	4	C	5	9	5	9		
F	2	D	5	9	7	7		
G	1	E,F	9	10	9	10		

An interesting case arises at the 1st vertex.



$$\left.\begin{array}{l}\text{LST of } B = 5-3=2\\ \text{LST of } A = 2-2=0\end{array}\right\}\text{LFT at vertex 1 is 0, which is the minimum of these two values}$$

Thus, in the backward pass we compute the latest start time and the latest finishing time of each activity according to the rules:

LFT of last vertex = EST of last vertex

LST = LFT – Duration

LFT = minimum of the LST's of the successive activities.

Float and critical path

Since an activity network generally involves a flow of activities through different paths of the network, one of these paths will become critical in the sense that any delay along this path will delay the whole project.

The critical path is the path where the Total Float is zero. Any delay along the critical path will delay the whole project.

Along other paths delay will be possible, and we call this delay "float".

But there are different senses in which delay is possible.

Total float (TF) is the amount of time by which an activity can be delayed without delaying the finishing time of the whole project.

For each activity the total float is

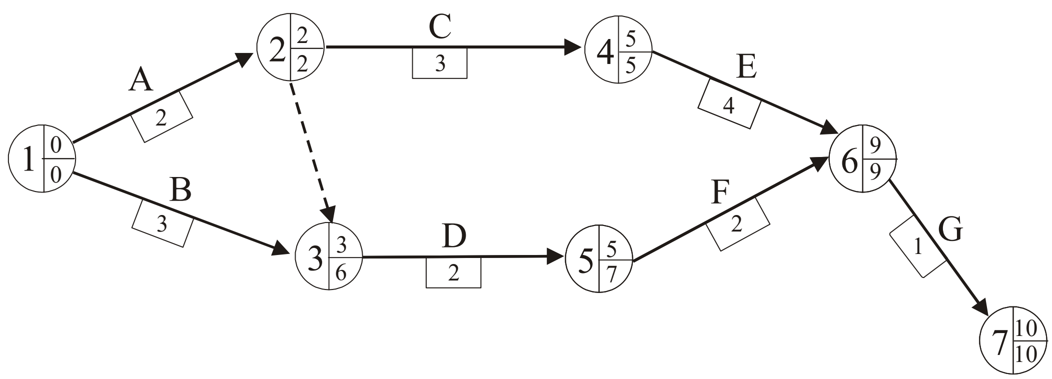
TF = LST – EST

This is the indication of the total number of time units that are free at that activity in the network as a whole. However, suppose there are two activities in the same part of a network where the total float is the same. The managers of both activities decide simultaneously to delay production on their activity by the value of the total float. Then the result is that the whole project is delay. The total float can only be used once. On the other hand, if each manager knew that the time they were delaying their activity by was *free float*, then they could use this free float without having to consult the other activity managers.

Free float is time that can be used up on an activity that is exclusive to that activity. It is defined by

FF = EFT – EST – Duration

These values are computed from the table.



Activity	Duration	Preceding Activity	EST	LFT	LST	EFT	TF	FF
A	2	-	0	2	0	2	0	0
B	3	-	0	5	2	3	2	0
C	3	A	2	5	2	5	0	0
D	2	A,B	3	7	5	5	2	0
E	4	C	5	9	5	9	0	0
F	2	D	5	9	7	7	2	0
G	1	E,F	9	10	9	10	0	0

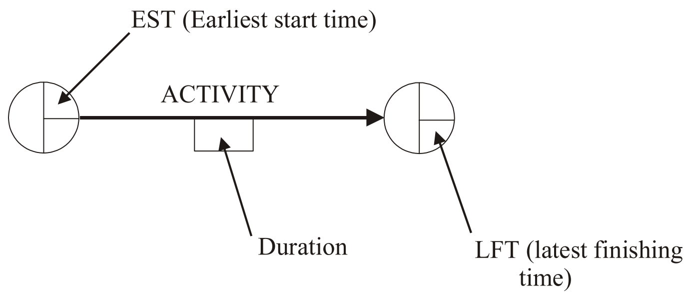
In this network there are three activities, *B, D* and *F*, where there is a total float of 2 units. If the total float is used at activity *B* then it cannot be used at activity *D* .

There are no points were there is purely free float – which is not surprising, because the whole example is not very complex, and one is only likely to get genuinely free float where there are many activities and so several non-critical paths.

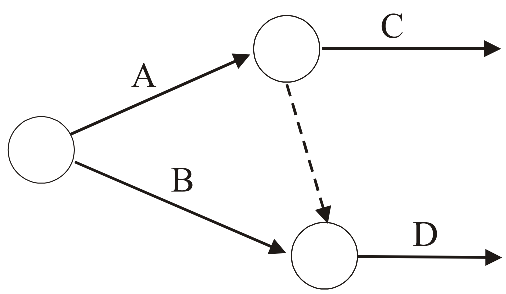
We conclude with a summary of the "rules" for critical path analysis.

Critical Path Analysis: summary

Activities



Dummy Activities



C is preceded by A
D is preceded by A & B

A dummy activity is represented by a dashed line of zero duration .

Forward Pass

EFT = Earliest finishing time   EFT = EST + Duration

EST = Earliest start time   EST = Max (EFT's of preceding activities)

Backward Pass

LFT = Latest finishing time   LFT of last vertex = EST of last vertex

LST = Latest start time   LST = LFT – duration

LFT = Min (LST's of successive activities)

Float

TF = LST – EST   TF = Total float

FF = EFT –EST –duration   FF = Free float