Critical path analysis

This section is about planning the order that jobs need to be done in so as the task is completed in the minimum amount of time.

For example planning a 3 course meal. There are certain tasks that must be done before others and other tasks that can be done at various times. We could make the dessert before we prepare the soup, but we would have to cook the main course before serving the food. Alternatively we might decide to make the dessert after the main course has been eaten.

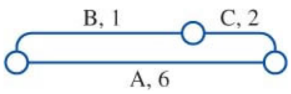
More complex activities require more planning and analysis. A network diagram can be used to represent the flow of activities.

Consider Frieda's morning schedule, where she needs to eat her cooked breakfast, download her email and read her email. The first two tasks take 6 minutes and 1 minute respectively, while the last takes 2 minutes. Frieda needs to complete all these tasks in 7 minutes. How might she accomplish this?

Clearly, she needs to be able to do some tasks simultaneously. Although this seems like a simple problem, let us look at what might happen each minute.

Time	Activity	Activity
1st minute		Download email
2nd minute	Eat breakfast	
3rd minute	Eat breakfast	
4th minute	Eat breakfast	
5th minute	Eat breakfast	Read email
6th minute	Eat breakfast	Read email
7th minute	Eat breakfast	

In the figure at below, the *edges* of our network represent the three activities of downloading (B), reading (C) and eating (A). The left *node* represents the start of all activity, the right node the end of all activity and the middle node indicates that activity B must occur before activity C can begin. In other words, activity B is the *immediate predecessor* of activity C.

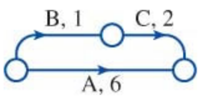


Another way of representing this information is in an activity chart.

Activity letter	Activity	Predecessor	Time (min)
A	Eat breakfast	—	6
B	Download email	—	1
C	Read email	B	2

This chart also shows that activity B (downloading) is the immediate predecessor of activity C (reading), and that activities B and C have no predecessors.

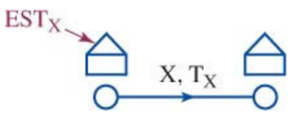
An alternative network diagram is shown at below.



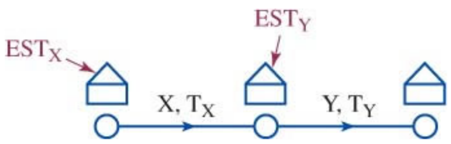
These activities can only be taken in a certain sequence so arrowheads are placed on the edges. These networks are called directed graphs or directed networks. You can see from the network diagram above that all of the tasks can be completed in a time of 6 minutes, but lets look at how that might be done so we can apply that to a more complex example.

Forward scanning

Forward scanning is about calculating the earliest start time (EST) that an activity can begin and then calculating the completion time for the whole project. The EST is determined by looking at all the previous activities, starting with the immediate predecessors and working back to the start of the project. An activity cannot begin before the completion of the predecessors. We record the EST on a network diagram as shown in the next example. The duration of the activity is T_x.



When a network includes two or more activities, the same labelling process is used.



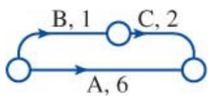
The purpose of the boxes beneath the triangles will be explained in a later section.

WORKED EXAMPLE 3

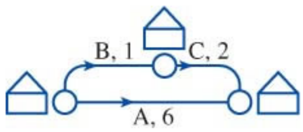
Use forward scanning to determine the earliest completion time for Frieda's initial three tasks.

THINK | WRITE/DISPLAY

1 Begin with the network diagram.

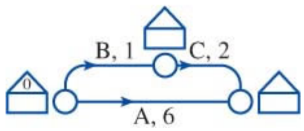


2 Add boxes and triangles near each of the nodes.

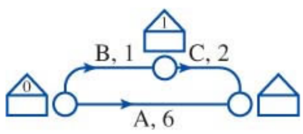


3 The earliest start time (EST) for each node is entered in the appropriate triangle. Nodes with no immediate predecessors are given the value of zero.

As activities B and A have no immediate predecessor then their earliest start time is zero.



4 Move to another node and enter the earliest start time (EST) in its triangle. In the case of activity C, it must wait one minute while its immediate predecessor, B, is completed.

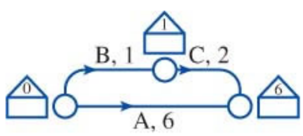


5 The last node's earliest start time is entered. When more than one edge joins at a node then the earliest start time is the largest value of the paths to this node. This is because all tasks along these paths must be completed before the job is finished.

There are two paths converging at the final node. The top path takes 3 minutes to complete and the bottom, 6 minutes. The larger value is entered in the triangle.

Path B–C = 1 + 2 = 3 minutes

Path A = 6 minutes



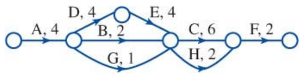
6 The earliest completion time is the value in the triangle next to the end node.

All tasks can be completed in 6 minutes.

Let us now extend Frieda's activity chart to a more complex set of activities for her morning routine.

Activity letter	Activity	Predecessor	Time (min)
A	Prepare breakfast	—	4
B	Cook breakfast	A	2
C	Eat breakfast	B, E, G	6
D	Have shower	A	4
E	Get dressed	D	4
F	Brush teeth	C, H	2
G	Download email	A	1
H	Read email	B, E, G	2
Total time			25

The network diagram for these activities is shown at below.

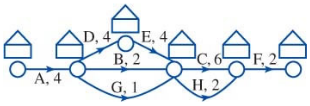


WORKED EXAMPLE 4

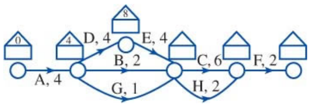
Using all the activities listed in Frieda's morning routine, find the earliest completion time and hence identify those tasks that may be delayed without extending the completion time.

THINK | WRITE/DRAW

1 Add the boxes and triangles to the directed network diagram.

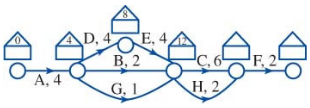


2 Begin forward scanning. The earliest start time (EST) for the first three nodes in the path can be entered immediately.



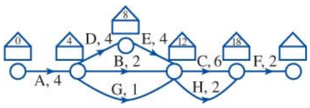
3 Calculate the time values for the paths to the fourth node. Enter the largest value (or longest time) into the appropriate triangle.

A–D–E = 4 + 4 + 4
= 12 minutes
A–B = 4 + 2
= 6 minutes
A–G = 4 + 1
= 5 minutes



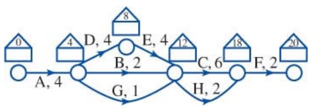
4 Repeat step 3 for the next node. Note that calculations begin by using the time from the previous node (12 minutes).

A–E–C = 12 + 6
= 18 minutes
A–E–H = 12 + 2
= 14 minutes



5 There is only one path to the last activity (F). Add its time requirement to that of the previous node (18 minutes).

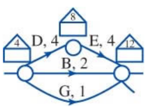
A–C–F = 18 + 2
= 20 minutes
Earliest completion time is 20 minutes.



6 The time in the last triangle indicates the earliest completion time.

Earliest completion time = 20 minutes

7 Identify sections of the network where there was a choice of paths. There are two such sections of the network. Examine the first one (the 4th node).

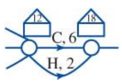


8 List and total the time for each path through this section of the network. The largest value indicates the path that cannot be delayed.

D–E = 4 + 4 = 8 minutes
B = 2 minutes
G = 1 minute
Paths B and G can be delayed.

9 Repeat step 8 for the next section identified in step 7.

C = 6 minutes
H = 2 minutes
H can be delayed.



The path through the network which follows those activities that cannot be delayed without causing the entire project to be delayed is called the *critical path*.

Therefore the critical path for the activities listed in Frieda's morning routine would be A–D–E–C–F. It is easily seen that this path takes the longest time (20 minutes).

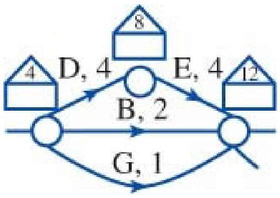
Jul 17-5:07 PM

Float time and latest start time

Float time is the difference in time between those paths that cannot be delayed and those that can. Quite often when completing projects with float time, activities can be delayed if there is a cost saving, otherwise they are done as soon as possible if appropriate. The latest start time (LST) is defined as the latest time they can start without delaying the project.

WORKED EXAMPLE 5

Work out the float time for activities B and G in Worked example 4, and hence identify the latest starting time for these activities.



THINK	DRAW
1 List the alternative paths for the section containing activities B and G and the times for these alternatives.	D–E = 4 + 4 = 8 minutes B = 2 minutes G = 1 minute
2 Subtract the smaller times separately from the maximum time.	Float time for activity B = 8 – 2 = 6 minutes Float time for activity G = 8 – 1 = 7 minutes
3 Look up the earliest completion time for the activity on the critical path and subtract the activity times.	D–E is on the critical path. Earliest completion time = 12 minutes Latest start time for activity B = 12 – 2 = 10 minutes Latest start time for activity G = 12 – 1 = 11 minutes

Drawing network diagrams

Let us now look at how networks are prepared from activity charts. As an example, we shall see how Frieda's morning schedule network was prepared.

WORKED EXAMPLE 6

From the activity chart below, prepare a network diagram of Frieda's morning schedule.

Activity letter	Activity	Predecessor	Time (min)
A	Prepare breakfast	—	4
B	Cook breakfast	A	2
C	Eat breakfast	B, E, G	6
D	Have shower	A	4
E	Get dressed	D	4
F	Brush teeth	C, H	2
G	Download email	A	1
H	Read email	B, E, G	2
Total time			25

THINK | WRITE/DRAW

1 Begin the diagram by drawing the starting node.

2 (a) Examine the table looking for activities that have no predecessors. There must be at least one of these. Why?

Activity A has no predecessors.

(b) This activity becomes the first edge and is labelled with its activity letter and arrowhead.

3 (a) List all activities for which A is the immediate predecessor.

Activity B has A as an immediate predecessor.

Activity D has A as an immediate predecessor.

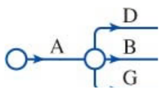
Activity G has A as an immediate predecessor.

(b) Add a node to the end of the edge for activity A.

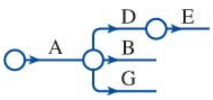
(c) Create one edge from this node for each of the listed activities.

Label these edges.

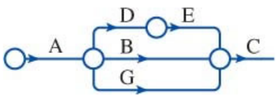
Note: The end node for each of these activities is not drawn until either you are certain that it is not the immediate predecessor of any later activities, or all activities have been completed.



4 Repeat step 3 for activity D. Since it is the *only* immediate predecessor of activity E, this can be added to the diagram. Otherwise, activity E could not be added yet.

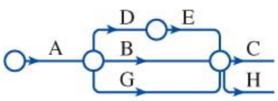


5 (a) Repeat step 3 for activities B and G. They have no activities for which they are the only predecessors. Since activity C is preceded by all of B, G and E, join all the edges at a single node.

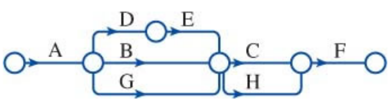


(b) Add activity C after this joining node. Note that activity H is also preceded by all of B, G and E but *not* by activity C.

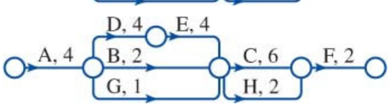
6 Determine whether activity C and H are independent of each other. Since they are independent, activity H starts from the same node as activity C.



7 The last activity is F, which has C and H as its immediate predecessors. Therefore join C and H with a node, then add an edge for F. Since F is the final activity, also add the end node.



8 Add the time required for each activity next to its letter.



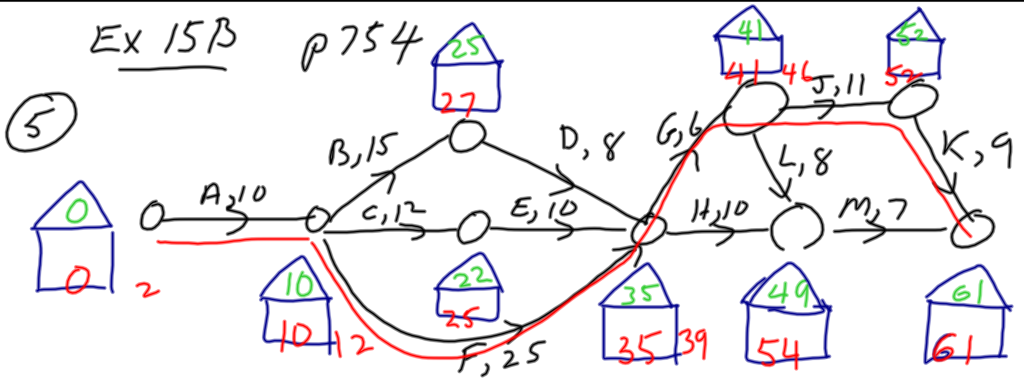
Ex 15 B page 754

Questions 1,2,3,4, 5, 8, 10, 11a,c, 12

Jul 17-5:27 PM

Ex 15B p754

(5)



(b) earliest completion time = 61

(8)

Activity	Immediate predecessor	Time
A	—	10
B	A	15
C	A	12
D	B	8
E	C	10
F	A	25
G	D, E, F	6
H	D, E, F	10
J	G	11
K	J	9
L	G	8
M	L, H	7

(10) (a) A - F - G - J - K

(b) B, C, D, E, H, L, M all have float time

Float time = LST − EST − time

(12)