

# 13

## Critical Path Analysis

### LEARNING OBJECTIVES

After studying this unit you will be able to :

- Explain the generic term network analysis
- Draw diagram & chart representing inter-relationships between various elements of the project and finally establishing critical path
- Identify Dummy Activities
- Identify earliest and latest event times of each event. Identify different types of Float Help in Decision Making about project times and resource allocation  
Calculating probability of completing network in a given time

### 13.1 Introduction

Planning, Scheduling and Controlling are three important functions of management. Planning involves the formulation of objectives and goals that are subsequently translated into specific plans and projects. Scheduling is concerned about the implementation of activities necessary to achieve the laid down plans. The function of control is to institute a mechanism that can trigger a warning signal if actual performance is deviating (in terms of time, cost and some other measures of effectiveness) from the plan. If such a deviation is unacceptable to the concerned manager, he will be required to take corrective action to bring performance in conformity with the plans. The PERT and CPM models are extremely useful for the purpose of planning, scheduling and controlling the progress and completion of large and complex projects or for carrying out the analysis of these three managerial functions. Before we describe the basic concepts used in the construction and analysis of these models, let us first understand the meaning of a project.

#### What is a project?

A *project* can be defined as a set of large number of activities or jobs that are performed in a certain sequence determined logically or technologically and it has to be completed within (i) a specified time, (ii) a specified cost and (iii) meeting the performance standards. Examples of a project from fairly diverse fields are given below:

1. Introducing a new product in the market.
2. Construction of a new bridge over a river or construction of a 25 storied building.
3. Executing a large and complex order on jobbing production.
4. Sending a spacecraft to the mars.

## 13.2 General Framework of PERT/CPM

A *network* is a graphical representation of a project, depicting the flow as well as the sequence of well-defined activities and events. Developed during the 1950s, both CPM (Critical Path Method) and PERT (Programme Evaluation and Review Technique) are network techniques/models. The network approach helps project managers in planning, scheduling and controlling. As a planning tool it helps the manager to estimate the requirements of resources viz., materials, equipment, manpower, cost and time for each activity or tasks of the project. This approach cannot make decisions by its own. It only provides additional information to executives to facilitate decision making process. Also it does not provide solution to every management problem. It certainly helps in identification of those activities, jobs or events which control the completion of the project.

### 13.2.1 Methodology of Critical Path Analysis

The working methodology of critical path analysis (CPA) which includes both CPM and PERT, consists of following five steps:

1. Analyse and break down the project in terms of specific activities and/ or events.
2. Determine the interdependence and sequence of specific activities and prepare a network.
3. Assign estimates of time, cost or both to all the activities of the network.
4. Identify the longest or critical path through the network.
5. Monitor, evaluate and control the progress of the project by re-planning, re-scheduling and re-assignment of resources.

The central task in the control aspect of these models is to identify the longest path through the network. **The longest path is the critical path because it equals the minimum time required to complete the project.** All other paths other than the critical path (i.e. non-critical or slack paths) offer flexibility in scheduling and transferring resources, because they take less time to complete than the critical path.

## 13.3 Advantages of Critical Path Analysis

There are a number of advantages in using critical path analysis.

1. It allows for a comprehensive view of the entire project. Because of the sequential and concurrent relationships, time scheduling becomes very effective. Identifying the critical activities keeps the executive alert and in a state of preparedness, with alternative plans ready in case these are needed. Breaking down the project into smaller components permits better and closer control.
2. Critical path analysis offers economical and effective system of control based on the principle of management by exception i.e. need for corrective action arises only in exceptional situations and in most of other cases, performance is in conformity with the plans.
3. It is a dynamic tool of management which calls for constant review, a reformulation of the network, and finding the current path of relevance and optimum resources allocation.

## 13.4 Fundamentals of a Critical Path Analysis (CPA) Network

### ❖ Activity

An activity is any portion of a project which consumes time or resources and has a definable beginning and ending. For example, "laying of pipe" is an activity requiring the use of resource mainly effort. Activity may involve labour, paper work, contractual negotiations, machinery operations, etc. Commonly used terms synonymous with "activity" are "task" and "job".

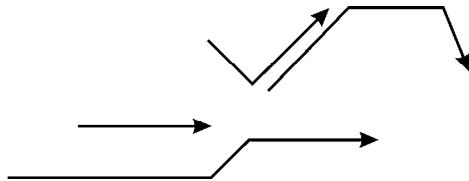


Figure 1

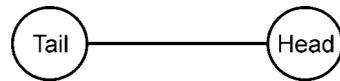


Figure 2

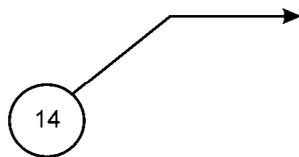


Figure 3

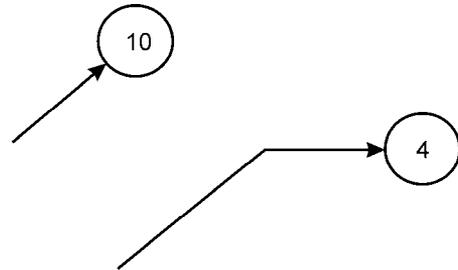


Figure 4

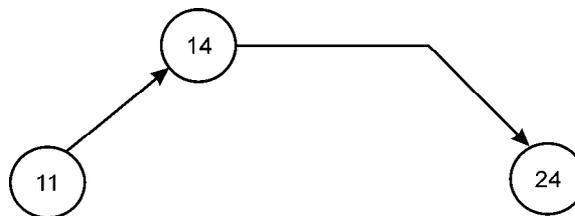


Figure 5

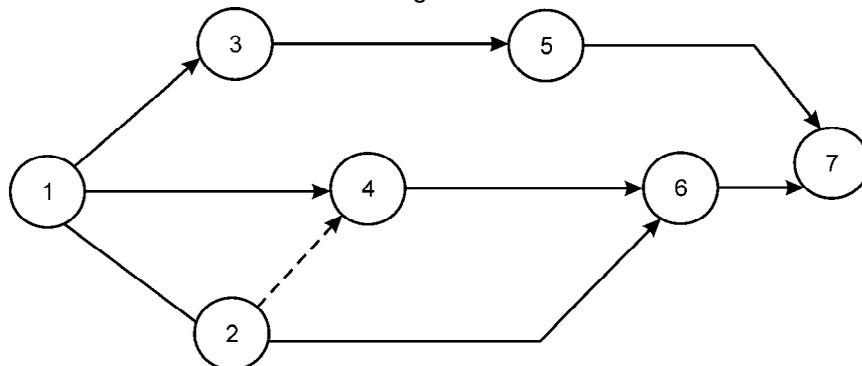


Figure 6

## 13.4 Advanced Management Accounting

Activities are graphically represented by arrows, usually with description and time estimates written along the arrows. The tail of the arrow portraying an activity represents the starting point of the activity and its head represents its completion. The arrow may be straight slanting, or bent but not broken (see figure-1). The arrow is not a vector and need not be drawn to scale.

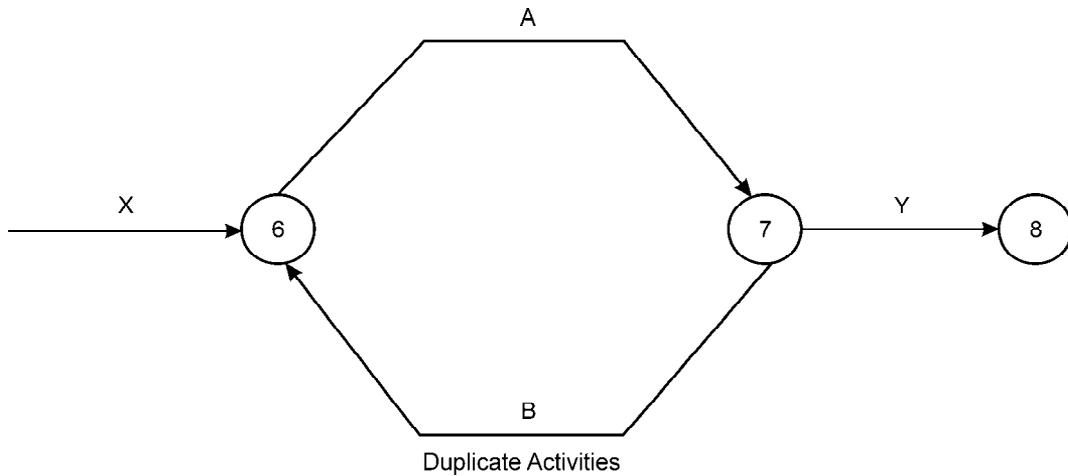


Figure 7

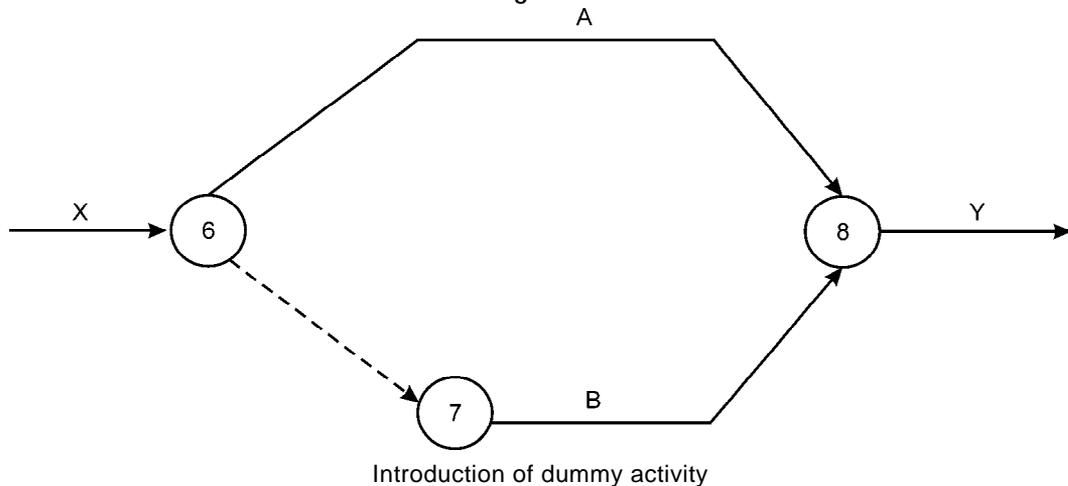


Figure 8

Normally an activity can be categorized into four activities:

- (i) **Predecessor activity:** The activity or activities which immediately come before another activity without any intervening activities are called Predecessor activities.
- (ii) **Successor activity:** The activity or activities which follow another activity without any intervening activities are called successor activity.
- (iii) **Concurrent activity:** Activities or tasks that can be carried out concurrently with another activity are called concurrent activity. An activity may be concurrent with one or more than one activities.

- (iv) **Dummy activity:** It is a hypothetical activity and does not consume any kind of resource. It is represented by dotted lines and is inserted in the network to clarify activity pattern under the following situations.
- to make activity/activities with common starting and finishing points distinguishable.
  - to identify and maintain the proper precedence relationship between activities that are not connected by events.
  - to bring all "loose ends" to a single initial and a single terminal event in each network using dummies, if necessary. When in a network diagram situation of duplicate activity arise, this situation is corrected by introducing a dummy activity.

#### ❖ Events

The beginning and ending points of an activity or a group of activities are called *events*. Synonyms of an event are "node" and "connectors". An event is often represented graphically by a numbered circle (see figure-2), although any geometric figure such as square, oval, rectangle etc. will serve the purpose. We shall, however, stick to the most commonly used convention for representing an event viz, the circle. A few examples of events are as follows: (i) Material procured, (ii) Design completed, (iii) Project started, (iv) Bricks laid, etc.

All activities in a network must commence from some event. Such events are called the *tail events* because they are connected to the tail of an activity. These are shown in figure 3. Similarly, all activities in a network must have terminal points called the *head event*, because it is at the head of an activity. These are shown in figure-4. Figure-5 depicts the tail and head events connected by arrows representing activities i.e. it depicts the dual role of an event. Event 14 is the head event for one activity and tail event for another.

In a network, symbol "*i*" is used for the tail event (also called preceding event) and "*j*" for the head event (or succeeding event) of an activity. The activity, then being *i-j*.

The events may be classified into three categories:

- Merge event:** If an event represents the joint completion of more than one activity i.e., a particular point (termed as event) where two or more activities complete is called merge event (See figure-9)
- Burst event:** If an event represents joint initiation i.e., a particular point (termed as event) from where more than one activities starts, is called Burst event (see figure-10).

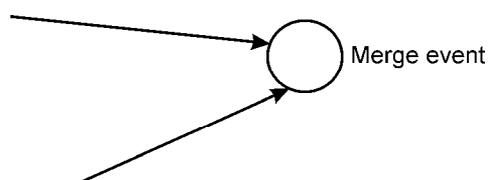


Figure 9

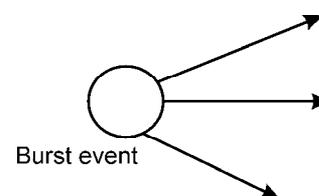


Figure 10

## 13.6 Advanced Management Accounting

---

- (iii) *Merge and Burst event*: It is an event from where more than one activity starts and more than one activity complete as well. In general language it is a starting point for some activities and ending point for some activities (see figure-11)

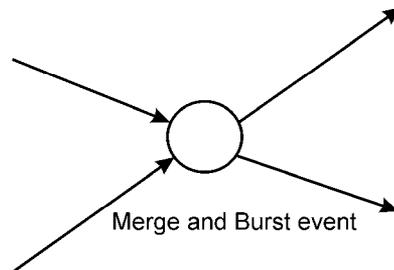


Figure 11

A network is, then, a graphical representation of a project plan, showing the inter-relationship of the various activities. Networks are also called *arrow diagrams* (see figure-6). When the results of time estimates and computations have been added to a network, it may be used as a project schedule.

**13.4.1 Conventions adopted in drawing networks:** There are two conventions normally adopted while drawing networks. In the early stages of network drawing, it is suggested that the conventions should be respected until sufficient experience has been gained to justify dropping them. These conventions are:

- (a) Time flows from left to right.
- (b) Head events always have a number higher than that of the tail events.

The above stated conventions allow activities to be referred uniquely by their tail and head event numbers, so that "activity 3-4" means only "the activity which starts from event 3 and proceeds to event 4"; it cannot mean "the activity which starts from event 4 and finishes at event 3".

**13.4.2 Graphical representation of events and activities:** Events are represented by numbers within circles. Activities are represented by arrows; the arrow-heads represent the completion of the activities. The length and orientation of the arrow are of no significance whatsoever (chosen only for the convenience of drawing). The activity of leaving place A and walking to place B can equally well be represented by figure-12.

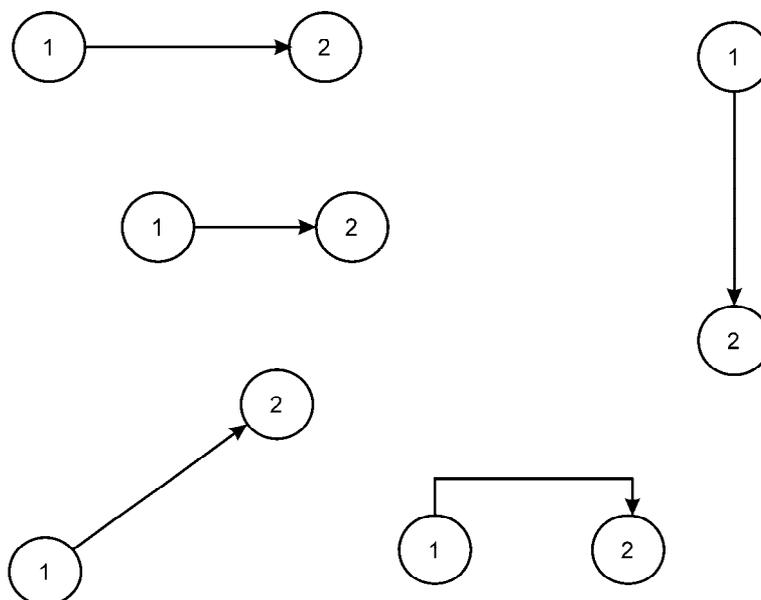


Figure 12

### 13.4.3 Fundamental properties governing the representation of events and activities:

The representation of events and activities is governed by one simple *dependency rule* which requires that an activity which depends upon another activity is shown to emerge from the head event of the activity upon which it depends and that only dependent activities are drawn in this way. Thus, if activity B depends upon activity A, then the two activities are drawn in figure-13.

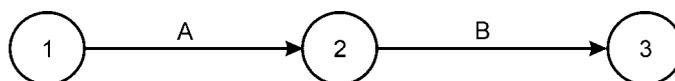


Figure 13

The dependency rule gives rise to two fundamental properties of events and activities:\*

1. An event cannot occur until all activities leading to it are complete.
2. No activity can start until its tail event is reached.

The above two properties can be combined into a single one, namely that "no activity may start until all previous activities in the same chain are completed."

**13.4.4 Logical sequencing and connection of activities:** A project entails several activities. The arrows are arranged to show the plan of logical sequence in which the activities of the project are to be accomplished. The sequence is ascertained for each activity by answering the following three queries viz:

- (i) Which activity or activities must be completed before the start of a particular activity?
- (ii) Which activity or activities should follow this?
- (iii) Which activities can be accomplished simultaneously?

Take an example of a pipe line project. Three activities have been identified for this project namely trenching, laying down of pipes and welding of pipes. To complete this project logical sequence is to be understood. Welding cannot be done until pipes are laid down and pipes cannot be laid down until the pipes are trenched. The logical sequence is like that first trenching second laying and lastly welding activity should be carried out. The activities will be as follows:

A → Trenching

B → Laying

C → Welding

The Network for the same is as follows

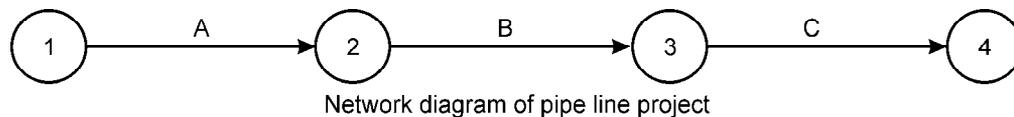


Figure 14

**13.4.5 Errors in logical sequencing:** Generally three types of errors in logical sequencing may arise while drawing a network diagram, particularly when it is a complicated one. These are known as *looping, dangling and redundancy*.

(1) **Looping:** Normally in a network, the arrow points from left to right. This convention is to be strictly adhered, as this would avoid illogical looping. Looping error is also known as Cycling error:



Figure 15

(2) **Dangling:** Activity which is not connected to any of the intermediate events or end event is called dangling activity. The situation represented by the following diagram is also at fault, since the activity represented by the dangling arrow 9-11 is undertaken with no result.

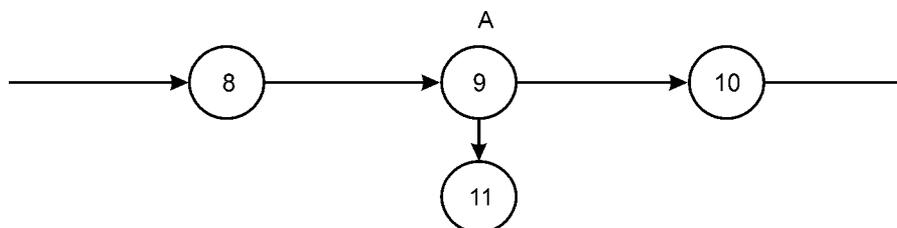


Figure 16

To overcome the problem arising due to dangling arrows, following rules may be adopted.

- (i) All events, except the first and the last, must have at least one activity entering and one activity leaving them, and
- (ii) All activities must start and finish with an event.
- (3) **Redundancy:** When dummy activities are inserted in a network diagram unnecessarily, this type of error is called error of redundancy. It is shown in the following figure:

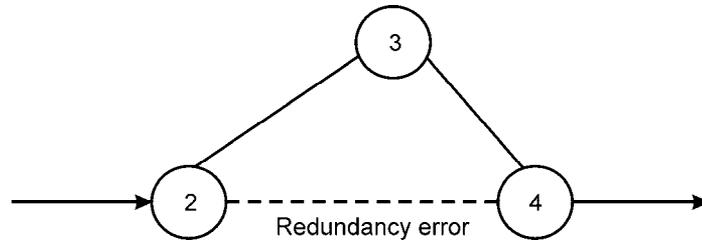


Figure 17

Figure-18 shows three cases for the following set of dependency relationships:

Activity C is dependent upon both A and B.

Activity D is dependent upon A alone.

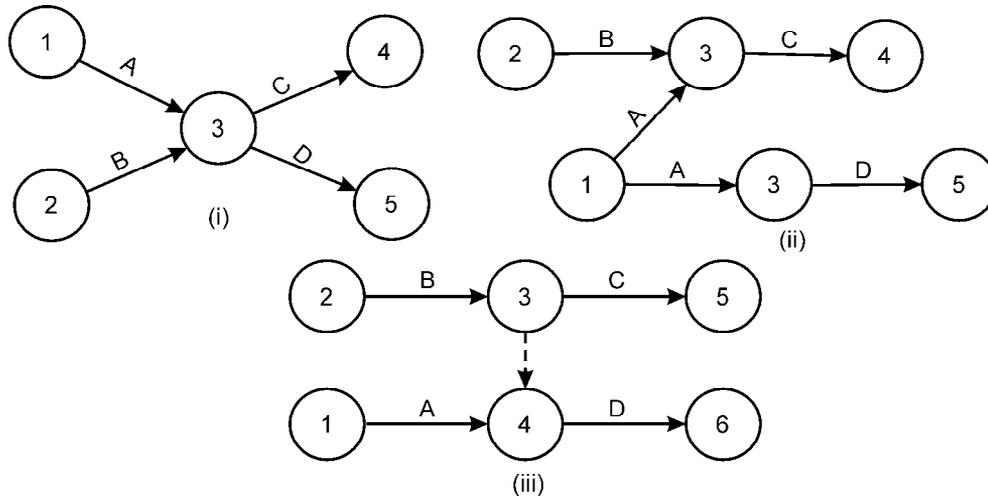


Figure 18

The first portrayal on (i) of figure-18 is clearly wrong since it shows D as dependent upon not only A but also B which is not desired. The other portrayal (ii) is also wrong since A is being shown twice and thus contravenes the fundamental axiom of network that there must be one arrow for each activity. The way out to this dilemma is the representation by means of the dummy activity. In the third portrayal of figure -18, C is dependent upon both A and B (via dummy) whereas D is dependent upon just A.

**13.4.6 Numbering the events:** The event numbers in a network should in some respect reflect their logical sequences. When a complicated network is drawn then the problem of assigning numbers to the events involved in the network arises. A rule devised by Delbert Ray

## 13.10 Advanced Management Accounting

Fulkerson, involving the following steps may be followed to resolve the problem of numbering the events.

- (i) An "initial" event is one which has arrow/arrows coming out of it and none of the arrow entering in it. In a network there will be only one such event. Call it "1".
- (ii) Delete all arrows coming out from the event 1. This will give us at least one more "initial event".
- (iii) Number these events as "2, 3...."
- (iv) Delete all emerging arrows from these numbered events which will create new initial events. Then follow step (iii).
- (v) Continue the above steps till last event is obtained which has no arrows coming out of it. Consider the numbering of events in the following figure.

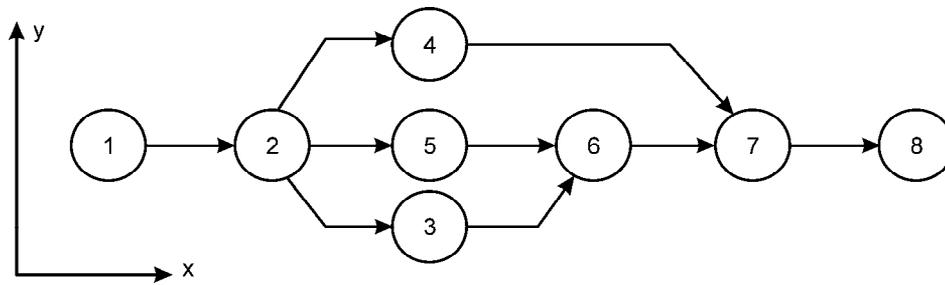


Figure 19

Here we proceed from left to right. The event with least x-co-ordinate is assigned the smallest integer, say 1. Other events are assigned progressively higher integers with regard to x-co-ordinate. If two or more events (4 and 5 above) have the same x-co-ordinate, the one towards arrow should have higher number.

Furthermore, it is not necessary, and in fact also not desirable to number the events consecutively. It would be a better scheme to number the events as 10, 20, 30, 40, 50, 60, 70, in the above diagram instead of 1,2,3,4,5,6,7. This affords insertion of more activities and events omitted by oversight or having become necessary in view of certain logic revisions.

It was mentioned earlier that it is desirable that all the activity arrows point from left to right. If the arrow is vertical it may point downwards or upwards.

For the sake of presentation it is to be recommended that activities emanating from one event or converging to another may make as great angles between themselves as possible.

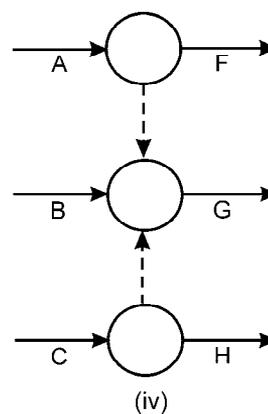
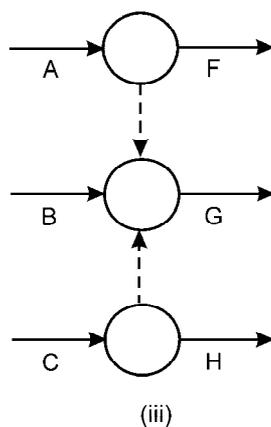
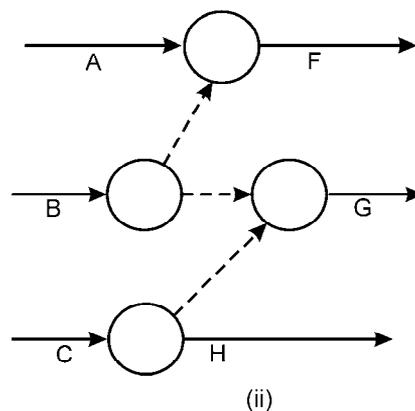
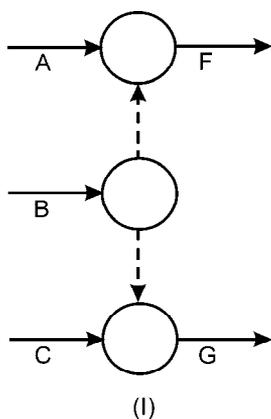
A few more conventions are given below:

- (i) Keep the arrow to the extreme right.
- (ii) As far as possible avoid drawing arrows that cross each other. Usually by suitable 'stretching' the network diagram it is possible to avoid this.
- (iii) Where, however, crossing is unavoidable, bridging may be done. This applies to dummies as well. Draw boldly a big network. Smaller ones are confusing. Use of pencil and rubber is recommended.

**Exercise:** Depict the following dependency relationships by means of network diagrams. The Alphabets stand for activities.

- (i) A and B control F; B and C control G.
- (ii) A and B control F; B controls G while C controls G and H.
- (iii) A controls F and G; B controls G; while C controls G and H.
- (iv) A controls F and G; B and C control G with H depending upon C.
- (v) F and G are controlled by A, G and H are controlled by B with H controlled by B and C.
- (vi) A controls F, G and H; B controls G and H with H controlled by C.

**Answer.** The required networks are given in figure-20



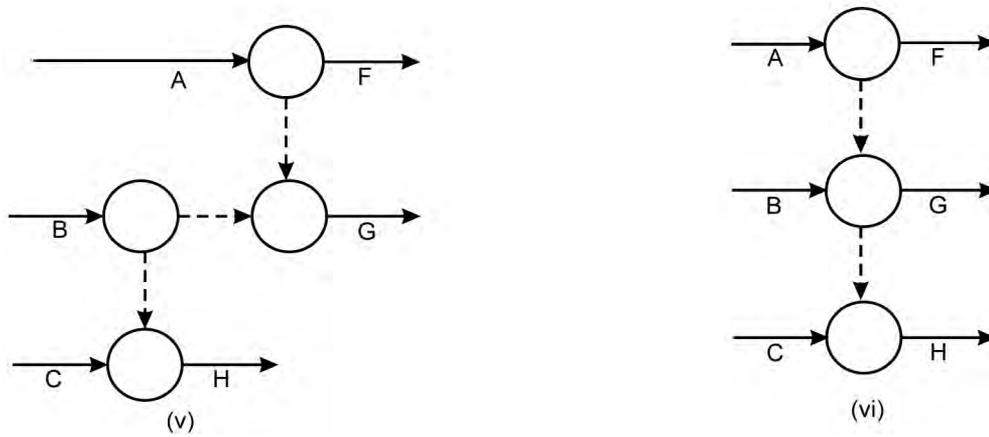


Figure 20

Exercise: Find out the superfluous (unnecessary) dummy activities in the networks below.

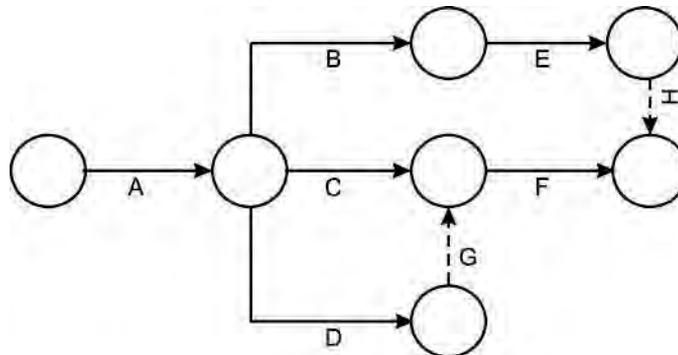


Figure 21

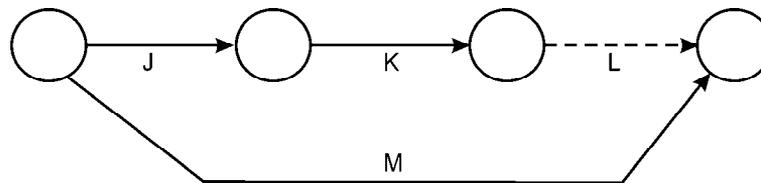


Figure 22

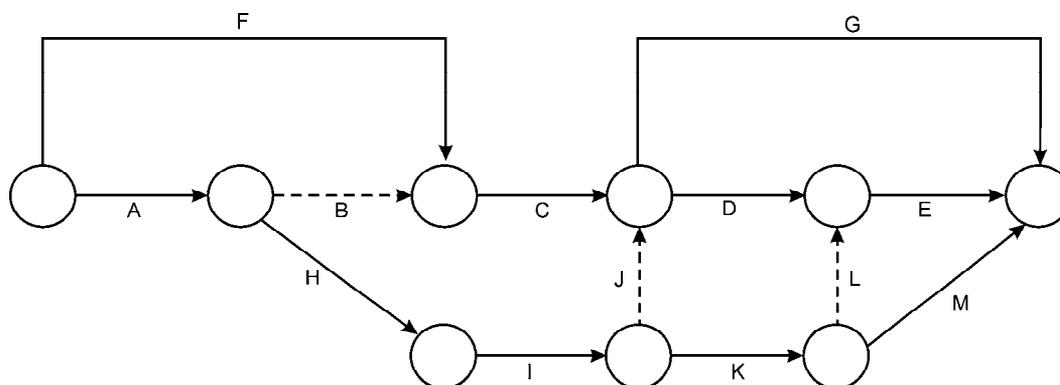


Figure 23

13.4.7 Basic steps involved in drawing a CPM/PERT network : Network is defined as a diagram representing the activities and events of a project, their sequence and inter-relationships.

The basic steps involved in drawing a network are:

- (i) Breaking up of the entire project into smaller systems known as tasks.
- (ii) For each task ascertain the activities and events to be performed.
- (iii) For each activity determine the preceding and succeeding activities.
- (iv) For each activity determine or estimate the time and other resources needed.
- (v) Draw a network depicting the assembly of tasks into a project.

13.4.8 Network Construction

*Illustration 1*

The activities involved in the computer installation process are detailed below. You are required to draw the network.

	<b>Activity</b>	<b>Predecessor Activities</b>
A.	<i>Physical Preparation</i>	<i>None</i>
B.	<i>Organisational Planning</i>	<i>None</i>
C.	<i>Personnel Selection</i>	<i>B</i>
D.	<i>Equipment Installation</i>	<i>A</i>
E.	<i>Personnel Training</i>	<i>C</i>
F.	<i>Detailed Systems Design</i>	<i>C</i>
G.	<i>File Conversion</i>	<i>F</i>
H.	<i>Establish Standards and Controls</i>	<i>F</i>
I.	<i>Programme Preparation</i>	<i>H</i>
J.	<i>Programme Testing</i>	<i>I</i>
K.	<i>Parallel Operations</i>	<i>D, E, G, J.</i>
L.	<i>Finalise Systems Documentation</i>	<i>I</i>
M.	<i>Follow up.</i>	<i>K, L</i>

## 13.14 Advanced Management Accounting

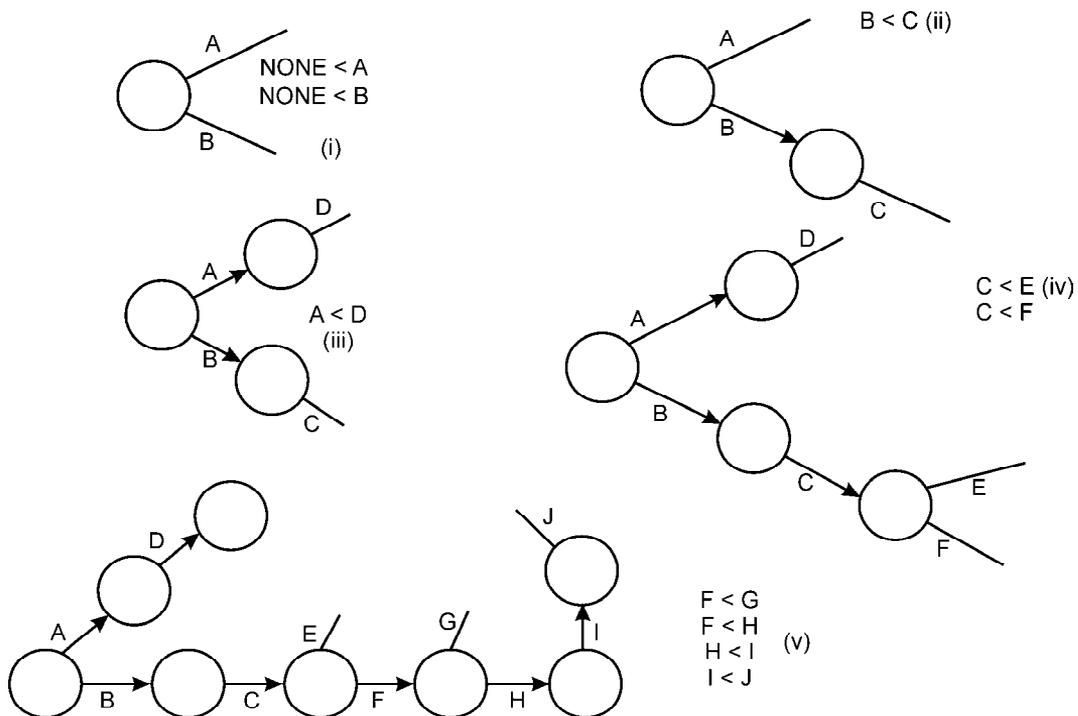
### Solution:

The network is shown in figure 24. The construction has been carried out step by step which is explained below. The symbol '<' is to be read as "precedes" in these figures.

- (i) Since A and B are preceded by no activity, they are shown emanating from the start event. We do not put arrows and the head events for A and B because they have been drawn as such only tentatively in the following step (s) their position may have to be changed.
- (ii) Here C is shown as dependent upon B. Again we neither show arrow and nor the head event of C because its position is tentative.
- (iii) D is shown as dependent upon A.
- (iv) We show here C preceding E and F.
- (v) F is shown preceding G and H, H precedes I and I precedes J.
- (vi) K is shown as dependent upon D, E, G and J, the latter four therefore, are made to converge on the same head event.
- (vii) L is shown as dependent upon I and M as dependent on K and L. Note carefully that K and L are the terminal activities and, therefore, converge on the last event.

About this point the student is cautioned since the beginners do tend to show the last activities dangling and do not converge them to the common last event.

Of course, the network would be constructed in just one diagram (as shown in Figure-24). We drew so many for explanation only. However, for a stylish presentation it is desirable that this one diagram is suitably sketched i.e. faired up in another diagram. The student may number the events as an exercise.



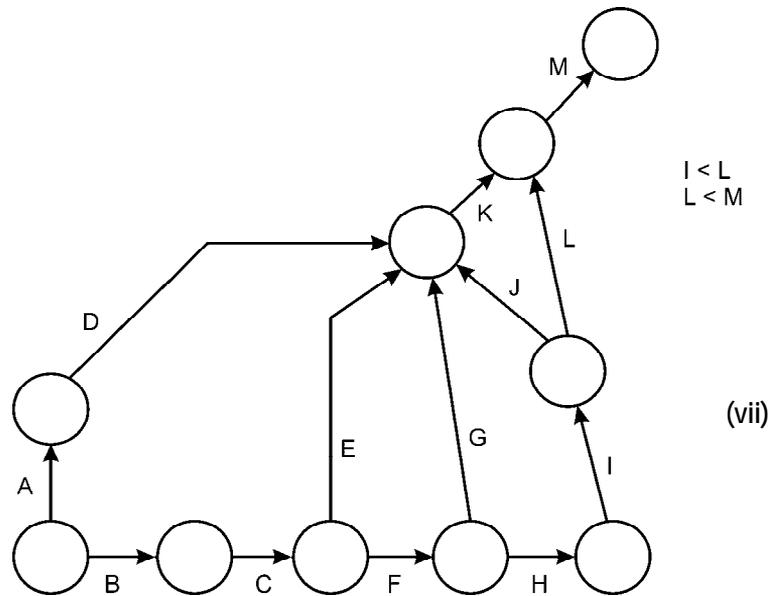
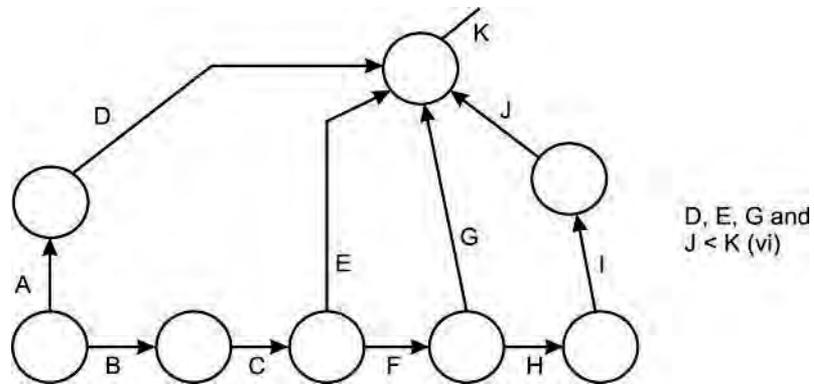


Figure 24

Illustration 2

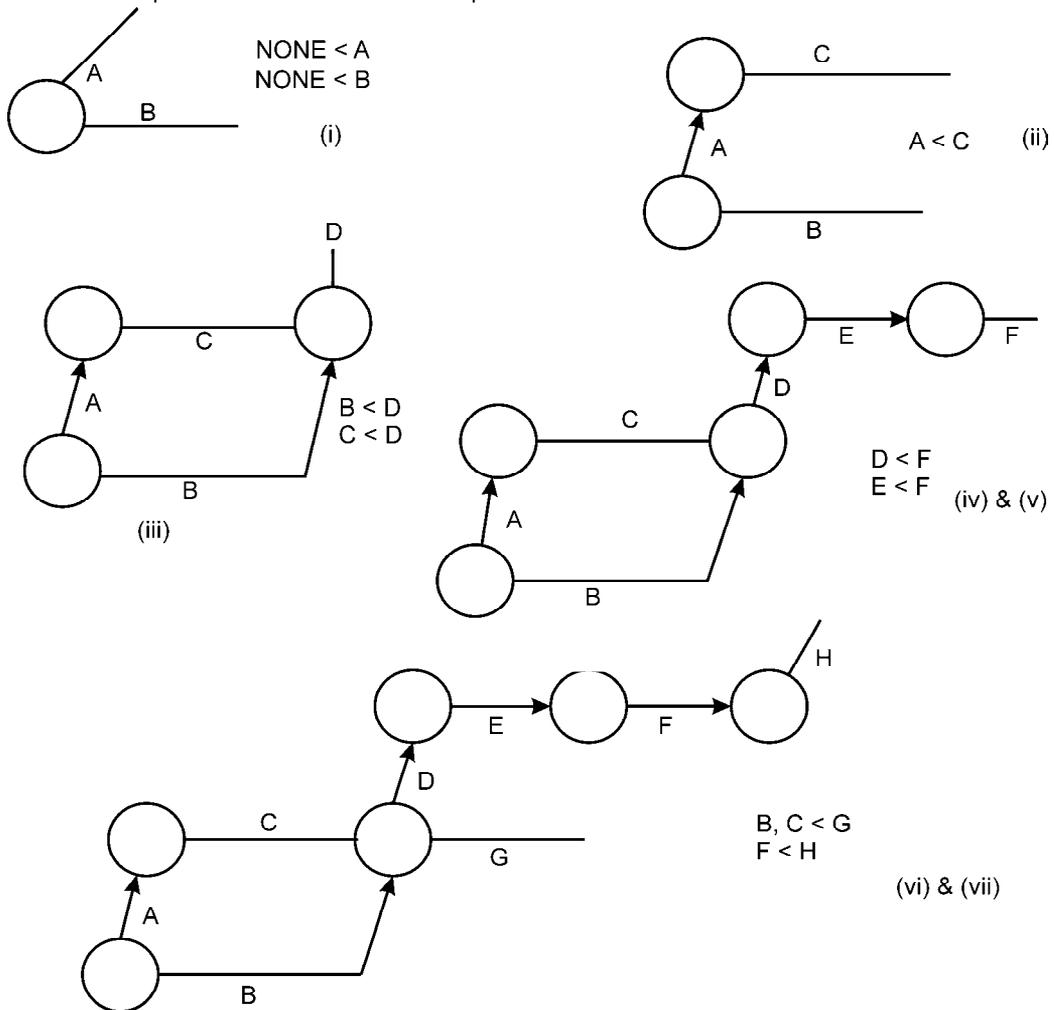
Activity	Preceded by
A.	None
B.	None
C.	A
D.	B, C
E.	D
F.	E
G.	B, C
H.	F

13.16 Advanced Management Accounting

<i>I.</i>	<i>F, G</i>
<i>J.</i>	<i>H, I</i>
<i>K.</i>	<i>B</i>
<i>L.</i>	<i>F, G, K</i>

**Solution:**

The network is shown in figure-25 step by step with dependency relationships that are incorporated at each step which the student should, by now, be able to follow. Please carefully review the steps where dummies are incorporated.



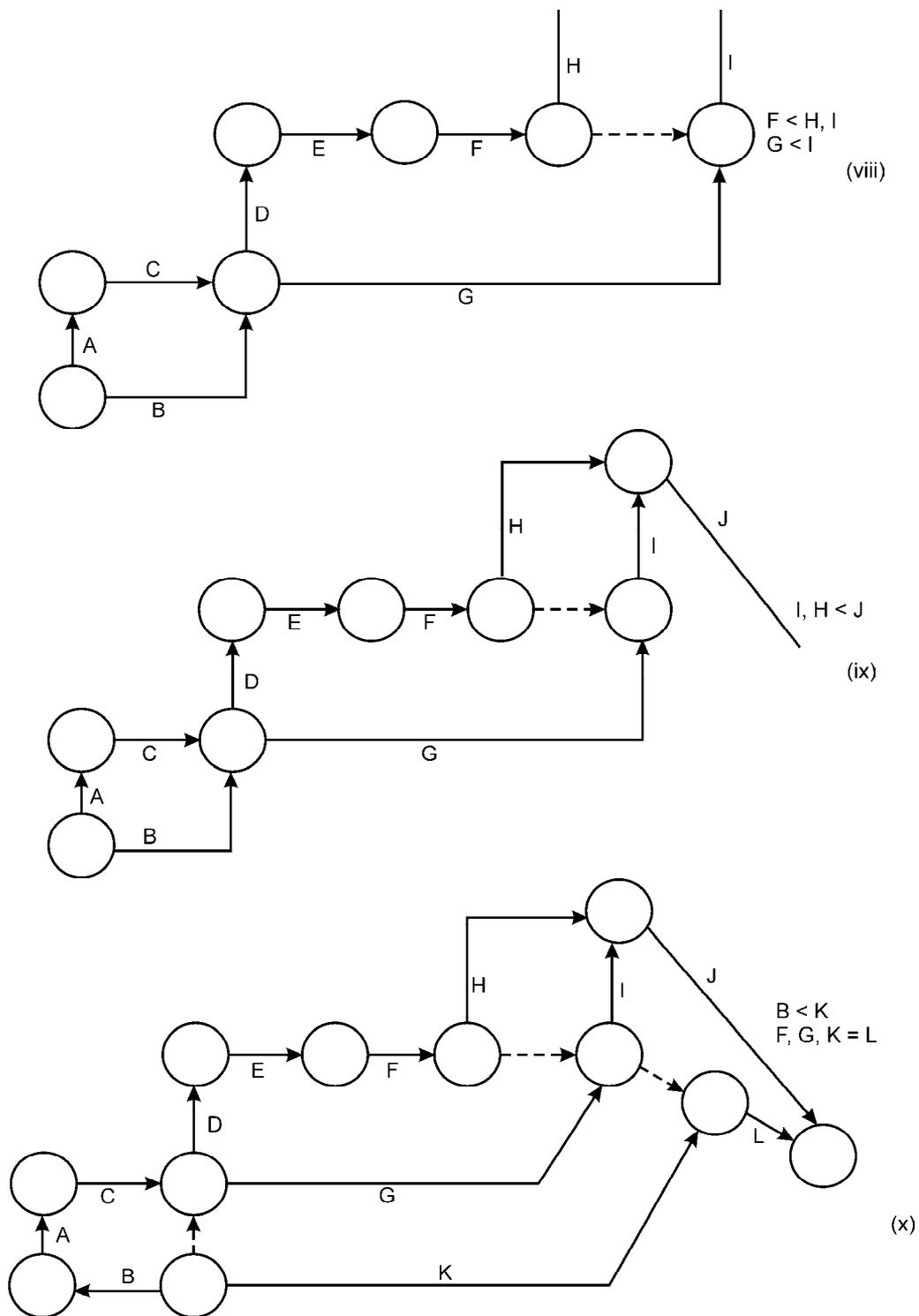


Figure 25

## 13.18 Advanced Management Accounting

---

**13.4.9 Concurrent activities:** Activities may not always be discrete i.e. they may be done in part allowing the subsequent activities to commence before the preceding activity is fully completed. Activities of this kind are to be frequently encountered in batch production. If, for example, a batch of 50 spindles is to be processed on two machines obviously it is not necessary to process all the items of the batch on the first machine and then transfer these to the next machine. A few items processed on the first machine may be transferred to the second machine before completion of the entire batch on the first machine. Since this is a matter of great practical importance we shall dwell upon it at a greater length. Such simultaneous or concurrent activities are to be encountered in sewage work e.g., trenching, laying pipe, welding pipe and back filling, all going on simultaneously with suitable lags on construction work.

### **Illustration 3**

*A batch of 4 axles is to be proceeded on the following three machines in this sequence: Lathe (L), Milling (M) and Grinding (G). Instead of first working on these 4 axles on lathe and then on milling and finally on grinding in sequence, it is desired to process the first axle on the lathe and as and when it is processed, it is taken up on milling and the 2nd axle on the lathe, and so on. In other words, each of the three activities L, M and G have been quartered for the sake of concurrent operations. You are required to draw the network.*

### **Solution**

The dependency relationships are sorted out hereunder:

<i>Quartered Activity</i>	<i>Preceded by</i>
L <sub>1</sub>	None
L <sub>2</sub>	L <sub>1</sub>
L <sub>3</sub>	L <sub>2</sub>
L <sub>4</sub>	L <sub>3</sub>
M <sub>1</sub>	L <sub>1</sub>
M <sub>2</sub>	L <sub>2</sub> , M <sub>1</sub>
M <sub>3</sub>	L <sub>3</sub> , M <sub>2</sub>
M <sub>4</sub>	L <sub>4</sub> , M <sub>3</sub>
G <sub>1</sub>	M <sub>1</sub>
G <sub>2</sub>	M <sub>2</sub> , G <sub>1</sub>
G <sub>3</sub>	M <sub>3</sub> , G <sub>2</sub>
G <sub>4</sub>	M <sub>4</sub> , G <sub>3</sub>

The network is now constructed below. (The student is urged to draw his own network. Mere understanding of our solution is not enough.)

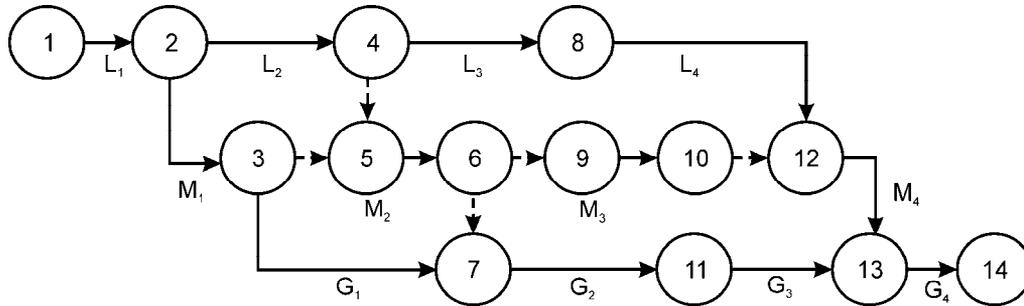


Figure 26

(N.B.: The concurrent activities so drawn are known as 'ladders' in the network jargon.)

**Illustration 4**

Draw a network diagram for the following data.

Task	Immediate Predecessor
A	—
B	—
C	B
D	B
E	B
F	E
G	A, D, C.

**Solution**

The required network is given below.

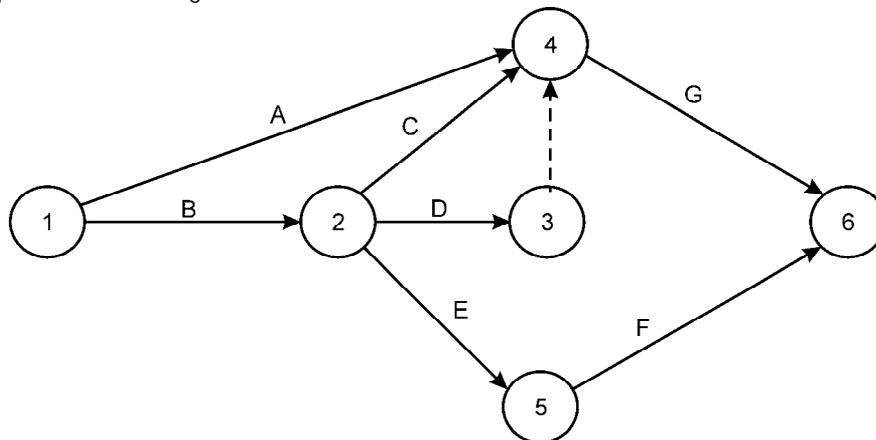


Figure 27

**Illustration–5**

Draw a network diagram for the following data.

Activity	Preceding activities
A	—
B	A
C	A
D	B
E	A
F	B, E
G	C
H	D, F
I	G
J	H, I

**Solution:** The required network is given below:

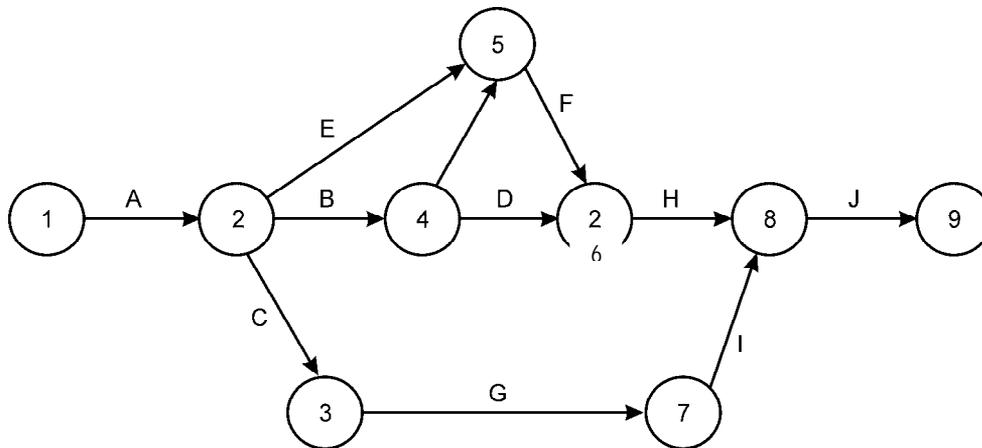


Figure 28

## 13.5 Critical Path Analysis

You have already been familiarised with the logic of the critical path analysis by way of introduction of this chapter.

The purpose of the analysis is two-fold: (i) **to find the critical path, i.e. the sequence of activities with the longest duration.** Once it is found it is marked in bold sequence of arrows on the network. For a simple network as of figure -29 the various sequences can be enumerated and the durations of activities encompassed by them simply added, to find the critical sequence. As stated earlier, one could indeed end up with more than one critical sequence; and (ii) **to find the float associated with each non-critical activity.**

**Systematic analysis:** The enumeration method would be too cumbersome computationally for any real life project, Even the computer would be hard pressed to proceed this way. There is a systematic way that cuts short the analysis time to manageable proportions.

It is accomplished by performing the following steps:

1. Calculate the time schedule for each activity. This represents the time by which an activity must begin and the time before which it must be completed. The time schedule data for each activity includes, the earliest start, earliest finish, the latest start, latest finish, and finally the float, which is the spare time associated with an activity.
2. Calculate the time schedule for the completion of entire project. This represents the scheduled date for the completion of the entire project and the probability of completing the project on or before the deadline.
3. Identify the critical activities. These activities are the ones which must be started and completed on schedule, or else the project is likely to be delayed.
4. Determine the critical path for the network. This path represents the critical activities which must be closely followed in order to complete the project on time.

**13.5.1 Scheduling computations:** After preparing the network diagram, we wish to know how long it will take to complete the project and also to identify the activities in the network that are to be placed under strict control. The basic scheduling computations involve a **forward** and then a **backward pass** through the network. The process of tracing the network from START to END is called forward pass, and from END to START is called backward pass. Based on a specified occurrence time for the initial network event, the forward pass computations give the earliest start and finish times for each activity. By the specification of the latest allowable occurrence time for the terminal network event, the backward pass computation will give the latest allowable start and finish times for each activity.

**13.5.2 Forward pass computations:** As stated above, the purpose of the forward pass is to compute the earliest start (EST) and finish time (EFT) for each activity. The EST time indicates the earliest time that a given activity can be scheduled. Earliest finish time for an activity indicates the time by which the activity can be completed, at the earliest. To compute these time estimates, we will first of all compute the earliest allowable occurrence time for various events of the network.

It is a convention to keep the earliest allowable occurrence time of the START event as zero.

To understand, how this time estimate for other events is computed, let us consider the following network diagram.

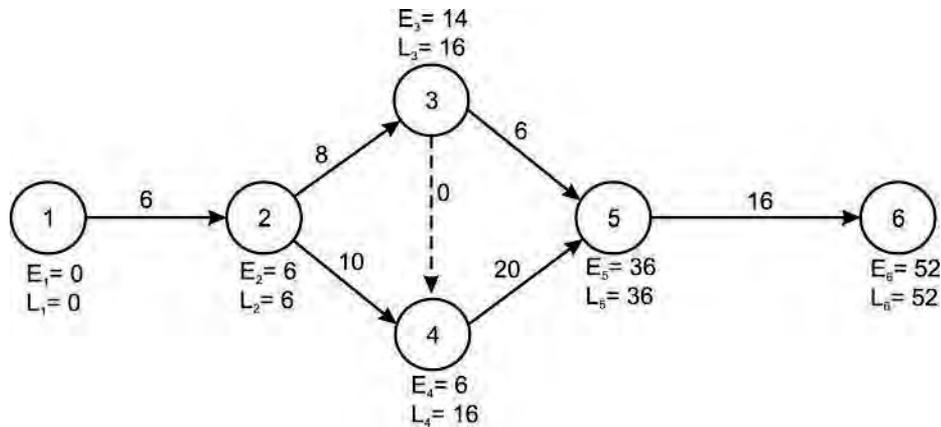


Figure 29

**Earliest allowable occurrence time of an event or Earliest Event Time:**

In the network shown above, event 1 stands for the beginning of the activity 1 – 2 and we can say that it occurs at the time zero i.e.  $E_1 = 0$ . Event 2 stands for the finish of the activity 1–2 thus event 2 can occur at the earliest time  $E_2$  which is computed as

$$E_2 = 0 + D_{12} = 0 + 6 = 6$$

Where,  $D_{12}$  stands for the duration of activity 1 – 2

Event 3 stands for finish of the activity 2–3 and its earliest time is

$$E_3 = E_2 + D_{23} = 6 + 8 = 14$$

The event 4 can occur either at the end of the activity 3 – 4 or at the finish of activity 2 – 4. In this case, there will be two time estimates as follows:

$$E_4 = E_3 + D_{34} = 14 + 0 = 14$$

$$E_4 = E_2 + D_{24} = 6 + 10 = 16$$

In case two or more time estimates exist for a particular event, then the time estimate with maximum value is retained as the earliest event time and other values are discarded. This maximum value represents the completion of all the activities ending at the event under consideration. In the above example, the earliest event time for event 4 will be 16.

A general rule can also be given here for determining the earliest event time as below:

$$E_j = \text{Max}(E_i + D_{ij})$$

Where,  $E_j$  is the earliest time for event  $j$ ,  $E_i$  is the earliest time for event  $i$ , and  $D_{ij}$  is the duration of the activity  $i-j$ .

❖ **Earliest start and earliest finish times of an activity**

After computing the earliest event time of various events, one can easily compute the earliest start and finish times of all the activities on the network. **The earliest start time of an activity**

is given by the earliest allowable occurrence time of the tail event of that activity. Thus, in our example, the earliest start time of the activity 1–2 will be given by the earliest time of the event 1 i.e. it will be 0. The earliest start time for the activities 2 – 3 and 2 – 4 will be given by the earliest time of event 2 which is equal to 6. The earliest time for the activities 3–4 and 3–5 will be 14 which is the earliest time for the event 3 and so on.

The earliest finish time of an activity will be simply equal to the earliest start time of the activity plus the duration of that activity. Hence, in our example, earliest finish time of activity 1–2 will be  $0+6 = 6$ , for activity 2–3, it will be  $6+8 = 14$  and for activity 2–4, it will be  $6+10 = 16$  and so on.

The complete computations for all the activities are shown in columns (3) and (4) of table -1 given on the next page.

**13.5.3 Backward pass computations:** The purpose of the backward pass is to compute the *latest start (LST) and finish times (LFT)* for each activity. These computations are precisely a "mirror image" of the forward pass computations. The term "latest allowable occurrence time of an event" (denoted by  $L_i$ ) is used in the sense that the project terminal event must occur on or before some arbitrary scheduled time. **Thus, the backward pass computations are started rolling back by arbitrarily specifying the latest allowable occurrence time for the project terminal event.** If no schedule date for the completion of the project is specified, then the convention of setting the latest allowable time for the terminal event equal to its earliest time, determined in the forward pass, is usually followed, i.e.  $L = E^*$  for the terminal event of the project. This convention is called the **zero slack convention**. **Following this, one can also interpret the latest allowable activity finish time (LFT) as the time to which the completion of an activity can be delayed without directly causing any increase in the total time to complete the project.**

To explain the computation, let us again consider the network diagram in figure-29. The terminal event is 6 so we set  $L_6 = E_6 = 52$  and we start rolling back. The latest allowable occurrence time for the events 5 and 4 are  $L_5 = 52 - 16 = 36$  and  $L_4 = 36 - 20 = 16$  respectively. It may be noted here that we can roll back to event 3 via activity 3-5 as well as activity 4–3. So there are two latest allowable occurrence times for the event 3 as given below;

$$L_3 = L_4 - D_{34} = 16 - 0 = 16$$

$$L_3 = L_5 - D_{35} = 36 - 6 = 30$$

We retain the minimum value as the latest occurrence time for the event 3 and ignore other values. Therefore, the latest allowable occurrence time for the event 3 is 16. Similarly

$$L_2 = L_3 - D_{23} = 16 - 8 = 8$$

$$\text{and } L_2 = L_4 - D_{24} = 16 - 10 = 6$$

The latest occurrence time for the event 2 is thus 6 and the latest occurrence time for the event 1 is equal to its earliest time i.e. zero.

In general, the latest allowable occurrence time of an event can be calculated by selecting an appropriate formula among the following two:

## 13.24 Advanced Management Accounting

$$L_j = L_j - D_{ij}$$

or  $L_j = \text{minimum } (L_j - D_{ij})$

The second formula is used for the event having two or more latest allowable occurrence time estimates.

### ❖ Latest start and latest finish times of an activity

After computing latest allowable occurrence time for various events, one can compute the latest start and latest finish times of an activity. **The latest finish time of an activity is equal to the latest allowable occurrence time of the head event of that activity.**

i.e.  $LFT(i-j) = L_j$

**The latest start time of an activity is equal to its latest finish time minus its duration.**

i.e.  $LST(i-j) = LFT(i-j) - D_{ij}$

These computations are shown in column (4) and column (6) of table-1 given below:

Table 1

Activity	Duration	START		FINISH	
		Earliest time	Latest time	Earliest time	Latest Time
(1)	(2)	(3)	(4)	(5)	(6)
1-2	6	0	6	0	6
2-3	8	6	14	8	16
2-4	10	6	16	6	16
3-4	0	14	14	16	16
3-5	6	14	20	30	36
4-5	20	16	36	26	36
5-6	16	36	52	36	52

**13.5.4 The critical path determination:** After having computed various time estimates, we are now interested in finding the critical path of the network. A network will consist of a number of paths. A path is a continuous series of activities through the network that leads from the initial event (or node) of the network to its terminal event. For finding the critical path, we list out all possible paths through a network along with their duration. In the network under consideration, various paths have been listed below.

Path	Length (in days)
1-2-3-5-6	36
1-2-4-5-6	52
1-2-3-4-5-6	50

❖ **Critical path:** A path in a project network is called critical if it is the longest path. The activities lying on the critical path are called the critical activities.

In the above example, the path 1–2–4–5–6 with the longest duration of 52 days is the critical path and the activities 1–2, 2–4, 4–5 and 5–6 are the critical activities.

**13.5.5 Calculation of Floats:** It may be observed that for every critical activity in a network, the earliest start and latest start time are the same. This is so since the critical activities cannot be scheduled later than their earliest schedule time without delaying the total project duration, they do not have any flexibility in scheduling. However, non-critical activities do have some flexibility i.e. these activities can be delayed for some time without affecting the project duration. This flexibility is termed as *slack* in case of an event and as *float* in case of an activity.

*Some people do not make any distinction between a slack and a float.*

❖ **Slack time for an event**

The slack time or slack of an event in a network is the difference between the latest event time and the earliest event time. Mathematically it may be calculated using the formula  $L_j - E_i$ . Where,  $L_j$  is the latest allowable occurrence time and  $E_i$  is the earliest allowable occurrence time of an event  $i$ .

❖ **Total float of an activity:** The total activity float is equal to the difference between the earliest and latest allowable start or finish times for the activity in question. Thus, for an activity (i-j), the total float is given by

$$TF_{ij} = LST - EST \text{ or } TF_{ij} = LFT - EFT$$

In other words, it is the difference between the maximum time available for the activity and the actual time it takes to complete. Thus, total float indicates the amount of time by which the actual completion of an activity can exceed its earliest expected completion time without causing any delay in the project duration.

❖ **Free Float:** It is defined as that portion of the total float within which an activity can be manipulated without affecting the float of the succeeding activities. *It can be determined by subtracting the head event slack from the total float of an activity.*

$$\text{i.e. } FF_{ij} = TF_{ij} - (\text{slack of event } j)$$

The free float indicates the value by which an activity in question can be delayed beyond the earliest starting point without affecting the earliest start, and therefore, the total float of the activities following it.

❖ **Independent float:** It is defined as that portion of the total float within which an activity can be delayed for start without affecting float of the preceding activities. *It is computed by subtracting the tail event slack from the free float of an activity.*

$$IF_{ij} = FF_{ij} - (\text{slack of event } i)$$

The independent float is always either equal to or less than the free float of an activity. If a negative value is obtained, the independent float is taken to be zero.

❖ **Interfering float:** Utilization of the float of an activity can affect the float of subsequent activities in the network. Thus, interfering float can be defined as that part of the total float which causes a reduction in the float of the successor activities. In other words, *it can be defined as the difference between the latest finish time of the activity under consideration and the earliest start time of the following activity, or zero, whichever is larger.* Thus, interfering float refers to that portion of the activity float which cannot be consumed without affecting adversely the float of the subsequent activity or activities.

**Illustration 6**

Activity	Duration
1-2	4 days
1-3	12 days
1-4	10 days
2-4	8 days
2-5	6 days
3-6	8 days
4-6	10 days
5-7	10 days
6-7	0 days
6-8	8 days
7-8	10 days
8-9	6 days

With the help of the activities given above draw a network. Determine its critical path, earliest start time, earliest finish time, latest start time, latest finish time, total float, free float and independent float.

**Solution**

The network based on the activities given in the example is as follows:

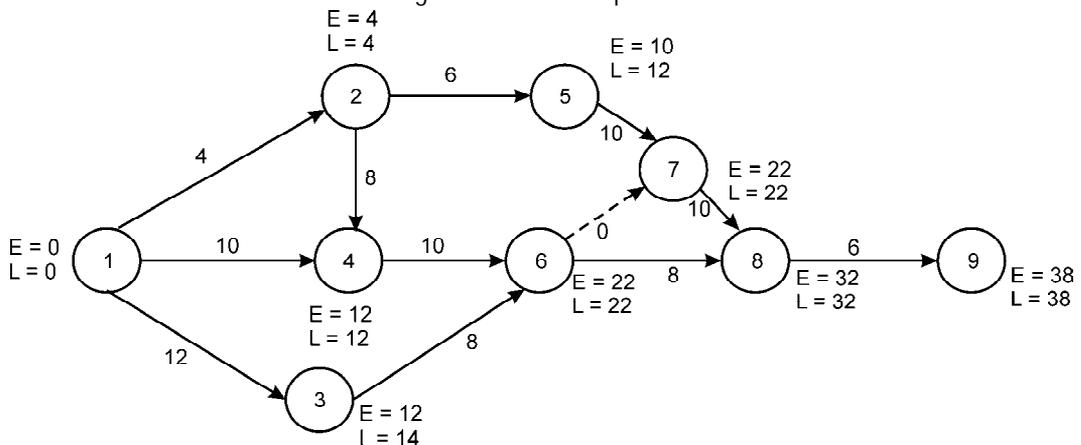


Figure - 30

The time estimates and floats are calculated as below:

Job or activity	Duration of an activity $D_{ij}$	EST (= $E_i$ )	EFT ( $E_i + D_{ij}$ )	LST ( $L_j - D_{ij}$ )	LFT (= $L_j$ )	Slack of		Total float (LST-EST)	Free float (Total Float - Slack of head event)	Independent (Free float - slack of tail event)
						tail event ( $L_i - E_i$ )	head event ( $L_j - E_j$ )			
1-2	4	0	4	0	4	0	0	0	0	0
1-3	12	0	12	2	14	0	2	2	0	0
1-4	10	0	10	2	12	0	0	2	2	2
2-4	8	4	12	4	12	0	0	0	0	0
2-5	6	4	10	6	12	0	2	2	0	0
3-6	8	12	20	14	22	2	0	2	2	0
4-6	10	12	22	12	22	0	0	0	0	0
5-7	10	10	20	12	22	2	0	2	2	0
6-7	0	22	22	22	22	0	0	0	0	0
6-8	8	22	30	24	32	0	0	2	2	2
7-8	10	22	32	22	32	0	0	0	0	0
8-9	6	32	38	32	38	0	0	0	0	0

Critical path is represented by 1-2-4-6-7-8-9. The project duration is 38 days.

(Note: In this table we have given detailed calculations merely for explanation.)

**Illustration 7**

Analyse the network below for the critical path and for different floats

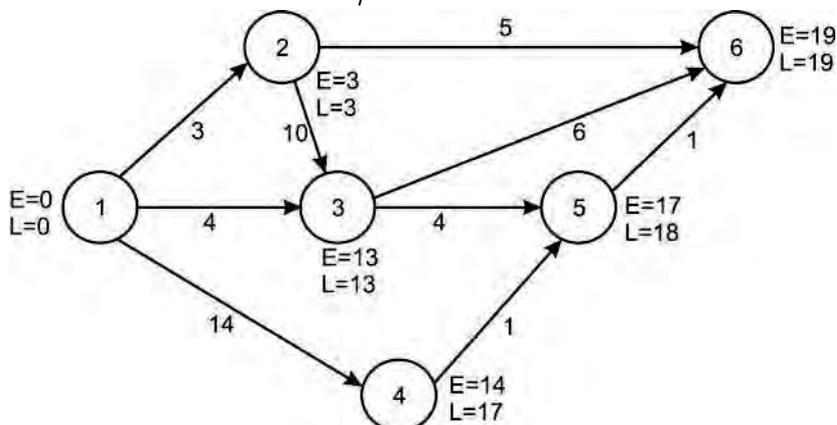


Figure 31

Solution

Network Analysis Table

Activity	Duration	Start		Finish		Float			
		E	L	E	L	Total	Free	Ind	Interfering
1-2	3	0	0	3	3	0	0	0	0
1-3	4	0	9	4	13	9	9	9	0
1-4	14	0	3	14	17	3	0	0	3
2-3	10	3	3	13	13	0	0	0	0
2-6	5	3	14	8	19	11	11	11	0
3-5	4	13	14	17	18	1	0	0	1
3-6	6	13	13	19	19	0	0	0	0
4-5	1	14	17	15	18	3	2	0	1
5-6	1	17	18	18	19	1	1	0	0

(N.B.: Activities must always be arranged in the i-j sequence.)

**Explanation:** Consider activity 1 – 3. The slack of head event 3 is  $13 - 13 = 0$ . Therefore, free float =  $9 - 0 = 9$ . Likewise slack of tail event 1 is  $0 - 0 = 0$ . Therefore independent float = 9.

**Illustration 8**

Analyse the network below for the critical path and calculate total float, free float and independent float for each activity.

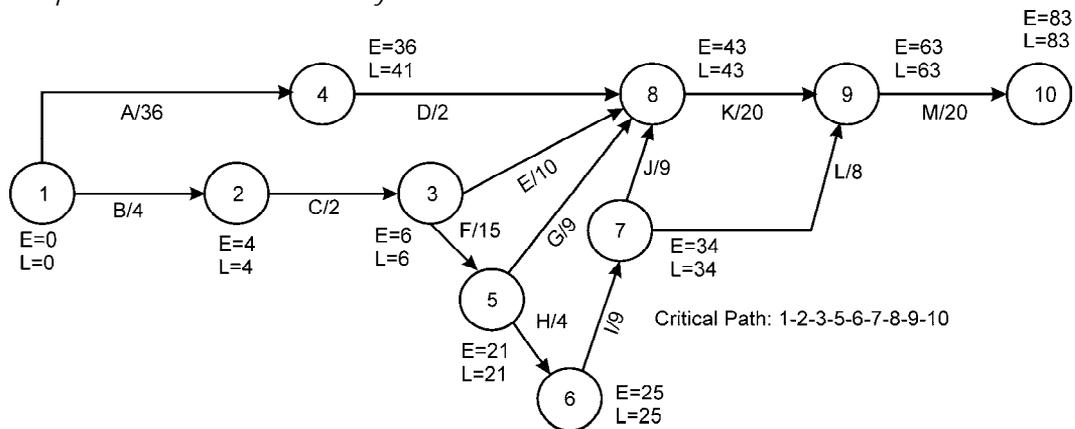


Figure 32

Solution

Network Analysis Table

i-j	D	Start		Finish		Floats		
		E	L	E	L	T	F	I
1-2	4	0	0	4	4	0	0	0
1-4	36	0	5	36	41	5	0	0
2-3	2	4	4	6	6	0	0	0
3-5	15	6	6	21	21	0	0	0
3-8	10	6	33	16	43	27	27	27
4-8	2	36	41	38	43	5	5	0
5-6	4	21	21	25	25	0	0	0
5-8	9	21	34	30	43	13	13	13
6-7	9	25	25	34	34	0	0	0
7-8	9	34	34	43	43	0	0	0
7-9	8	34	55	42	63	21	21	21
8-9	20	43	43	63	63	0	0	0
9-10	20	63	63	83	83	0	0	0

Critical Path: BCFHIJKM = 83 weeks.

### 13.6 Distinction Between PERT & CPM

PERT	CPM
(i) PERT stands for Program Evaluation Review Technique	(i) CPM stands for Critical Path Method
(ii) It is a Probabilistic tool using three estimates of duration	(ii) It is a deterministic tool, uses only single estimate of duration
(iii) It is a basically a tool for planning	(iii) CPM can be used to control both time and cash.
(iv) PERT is more suitable for R & D related projects.	(iv) CPM is best suited for routine projects/ works where time and cost and accurately.
(v) PERT is event oriented	(v) CPM is activity oriented.

## Summary

- Two most commonly used network Models
  - ✓ Critical Path Method (CPM)
  - ✓ Programme Evaluation Review Technique (PERT)
- Construction of Network diagrams
- Calculation of Critical Path and Project Completion time together with Floats, earliest start and finishing times
- Application of technique to crash cost problems
- Critical path should be longest route through the network
- Resource schedule problems and use of bar charts
- Identifying the project duration
- Cheapest critical activities to crash and not reduce project completion time by reducing non-critical activities