PERT/CPM

Project Scheduling

INTRODUCTION

- Schedule converts action plan into operating time table
- Basis for monitoring and controlling project
- Scheduling more important in projects than in production, because of unique nature
- Sometimes customer specified/approved requirement-e.g: JKR projects
- Based on Work Breakdown Structure (WBS)

NETWORK

- Graphical portrayal of activities and event
- Shows dependency relationships between tasks/activities in a project
- Clearly shows tasks that must precede (precedence) or follow (succeeding) other tasks in a logical manner
- Clear representation of plan – a powerful tool for planning and controlling project

EXAMPLE OF SIMPLE NETWORK – SURVEY

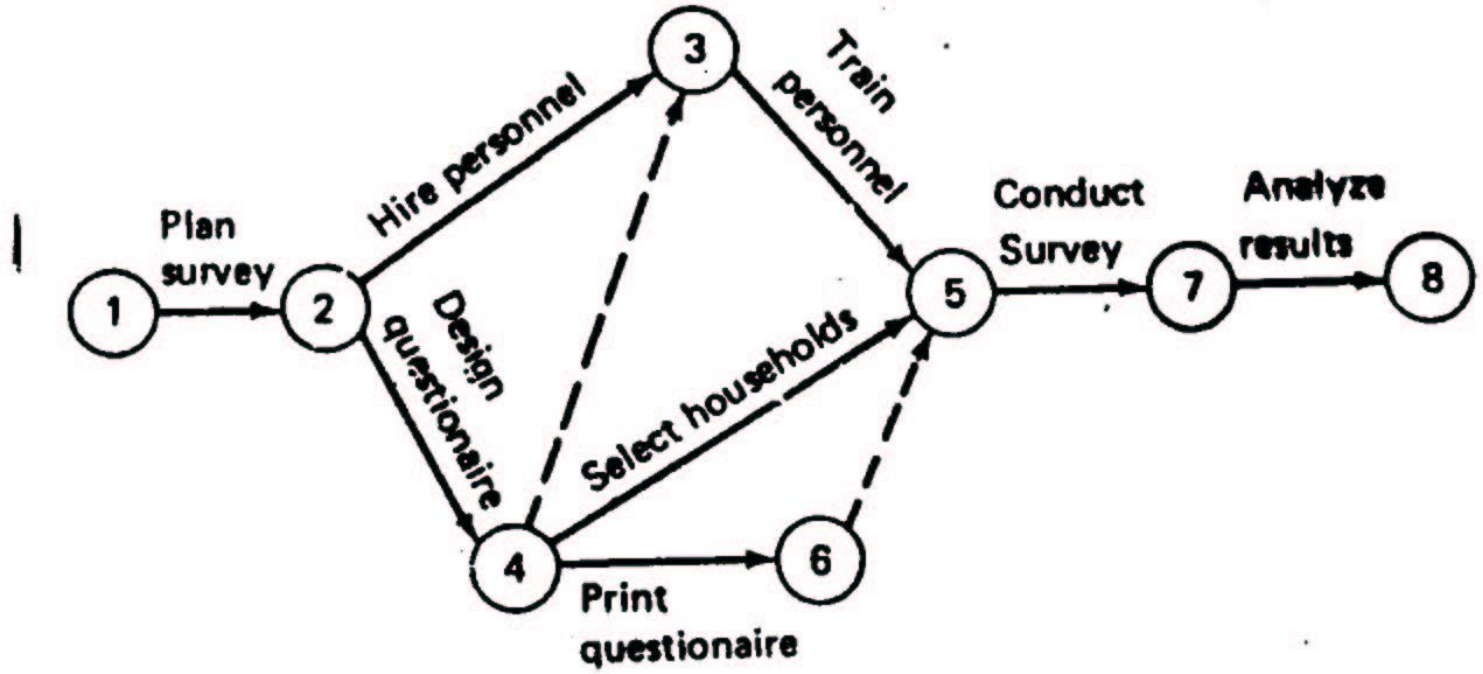


Figure 2-18

EXAMPLE OF NETWORK – MORE COMPLEX

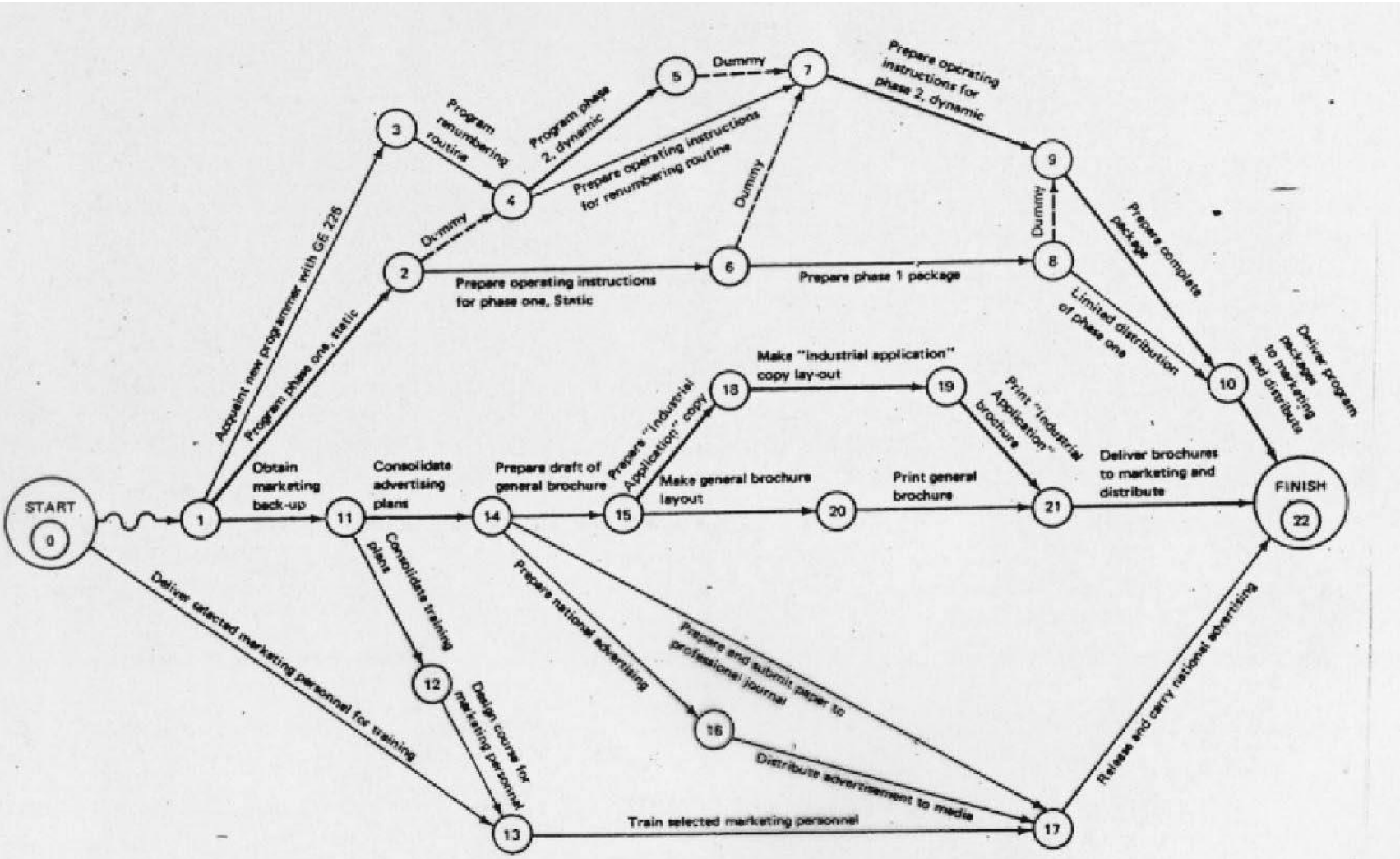


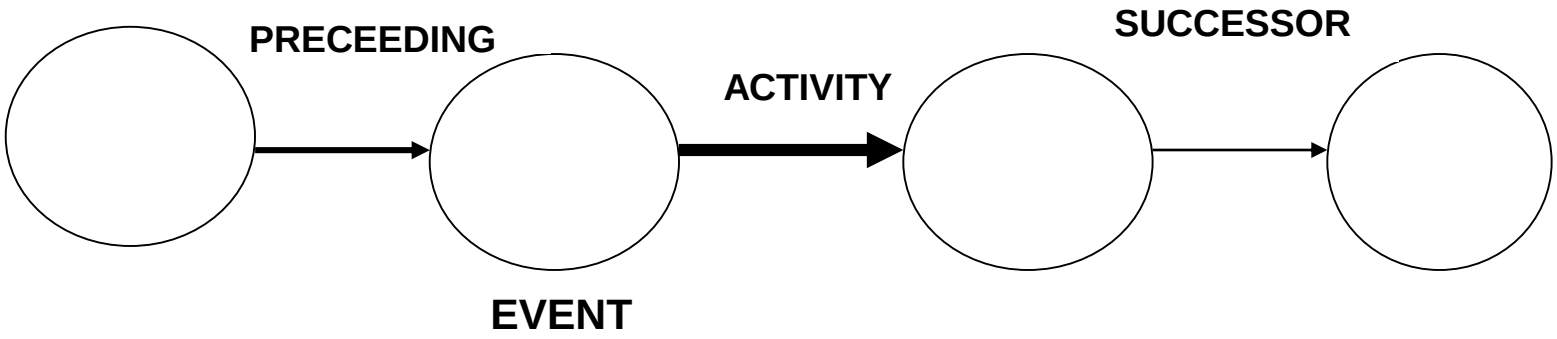
Figure 2-19 Network for the development and marketing of a new computer program.

PERT DIAGRAMS

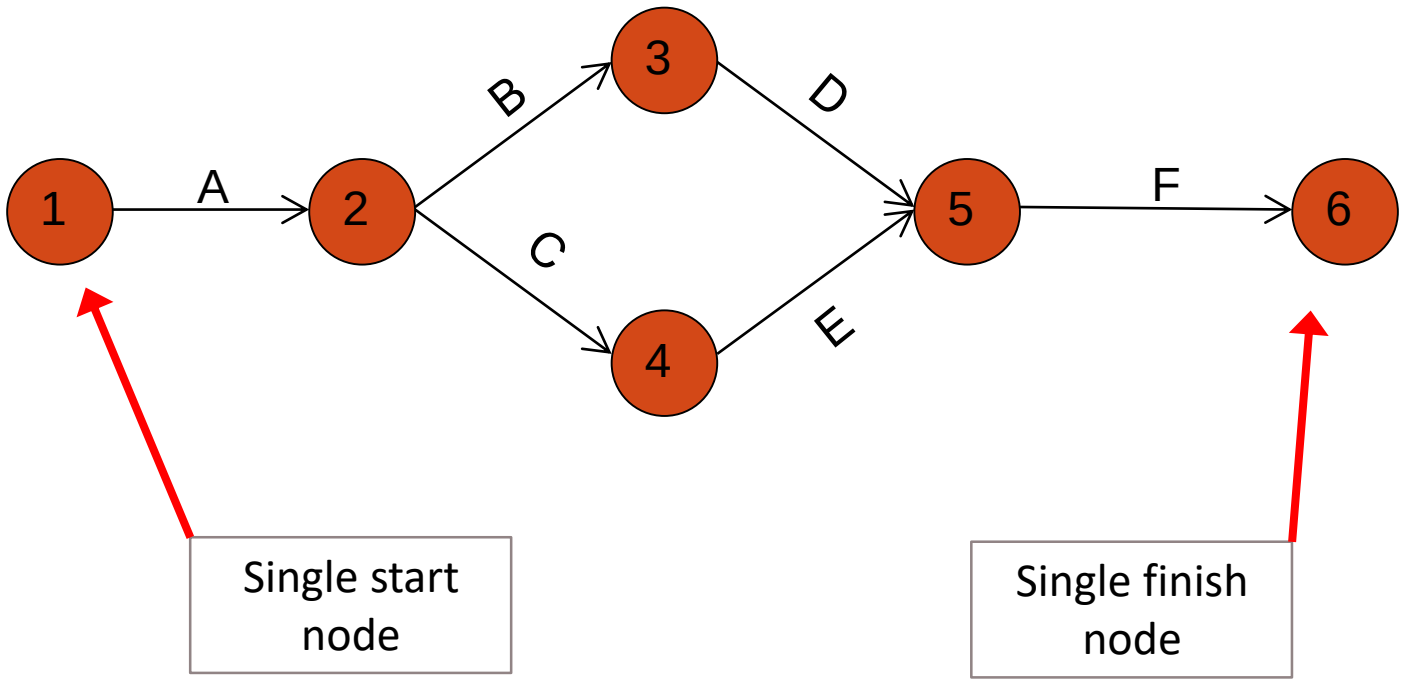
- Program Evaluation and Review Technique
- It is a network model that allows for randomness in activity completion times.
- Tool used to control the length of projects.
- PERT was developed in the late 1950's for the US Navy's Polaris Project.
- First used as a management tool for military projects
- Adapted as an educational tool for business managers
- It has the potential to reduce both the time and cost required to complete a project.

DEFINITION OF TERMS IN A NETWORK

- **Activity** : any portions of project (tasks) which required by project, uses up resource and consumes time – may involve labor, paper work, contractual negotiations, machinery operations Activity on Arrow (AOA) showed as arrow, AON – Activity on Node
- **Event :** beginning or ending points of one or more activities, instantaneous point in time, also called 'nodes'
- **Network** : Combination of all project activities and the events



PERT DIAGRAMS



PERT DIAGRAMS

- Activity-on-node diagrams:
  - Maybe more than one single start and end node
  - Nodes represent activities
  - Arrows indicate precedence

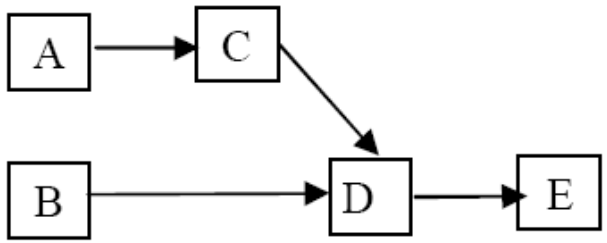
- Activity-on-arrow diagrams:
  - One single start and one single end node
  - Arrows represent activities
  - Nodes indicate beginning/end of activities

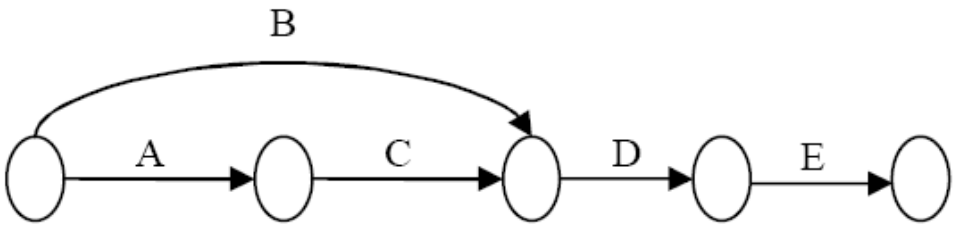
PERT DIAGRAMS

Example of PERT diagrams:

Task	Precedence
A	
B	
C	A
D	B,C
E	D

Activity-on-node

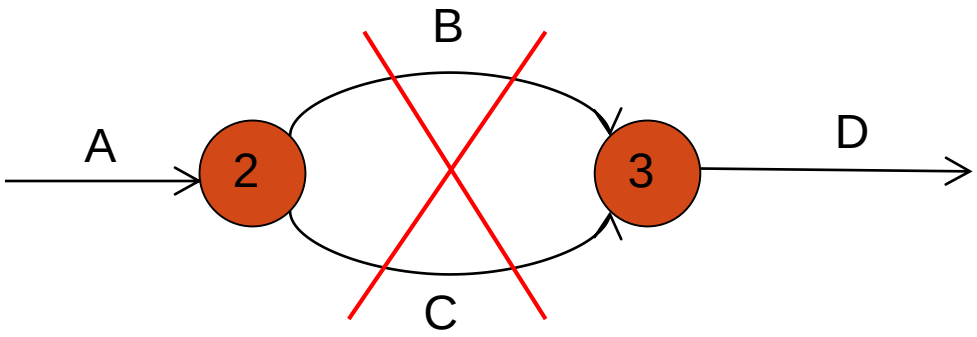




Activity-on-arrow

PERT DIAGRAMS

- Some basic rules for Activity –on-arrow:
  - Tasks are represented as arrows
  - Nodes represent the start and finish points of tasks
  - There is only one overall start node
  - There is only one overall finish node
  - Two tasks cannot share the same start and end node.



Tasks B & C share the same start and end node

EXAMPLE 1- A SIMPLE NETWORK

Consider the list of four activities for making a simple product:

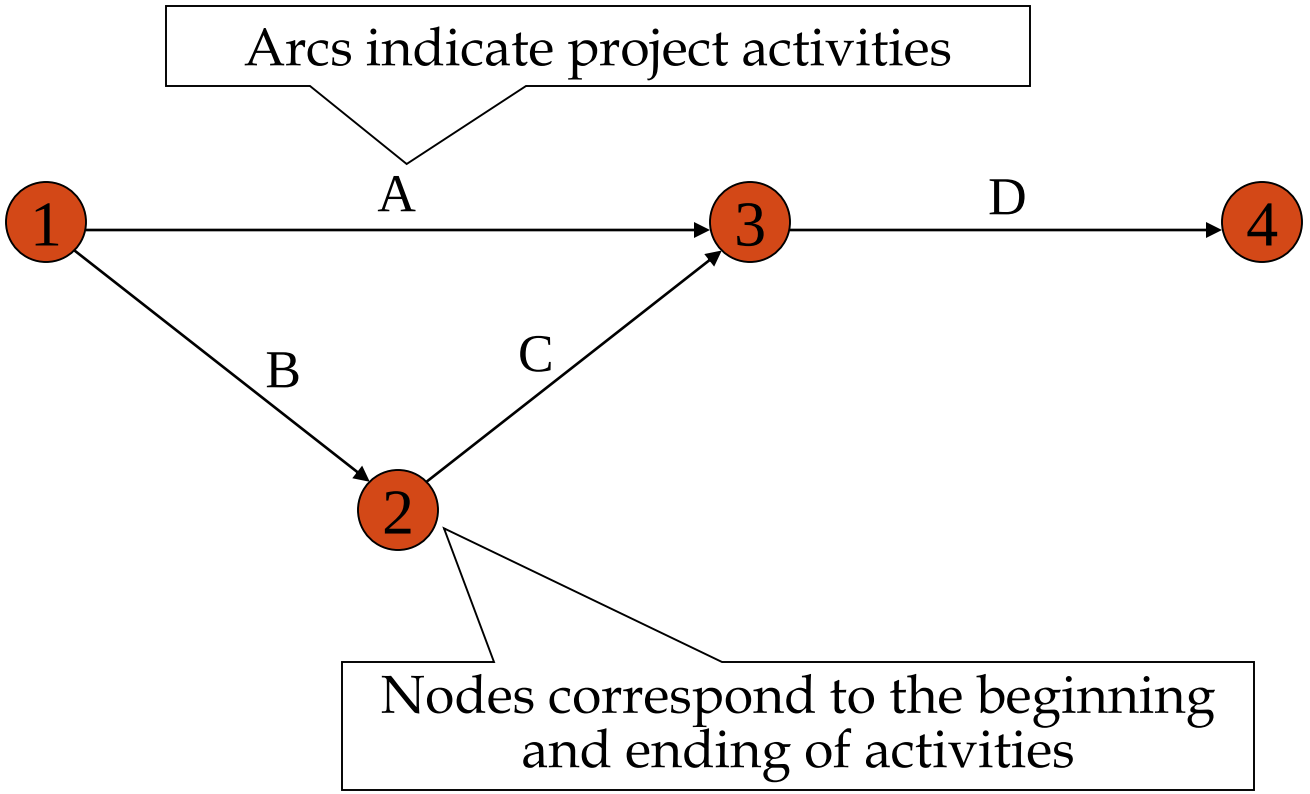
Activity	Description	Immediate predecessors
A	Buy Plastic Body	-
B	Design Component	-
C	Make Component	B
D	Assemble product	A,C

Immediate predecessors for a particular activity are the activities that, when completed, enable the start of the activity in question.

SEQUENCE OF ACTIVITIES

- Can start work on activities A and B anytime, since neither of these activities depends upon the completion of prior activities.
- Activity C cannot be started until activity B has been completed
- Activity D cannot be started until both activities A and C have been completed.
- The graphical representation (next slide) is referred to as the PERT/CPM network

NETWORK OF FOUR ACTIVITIES



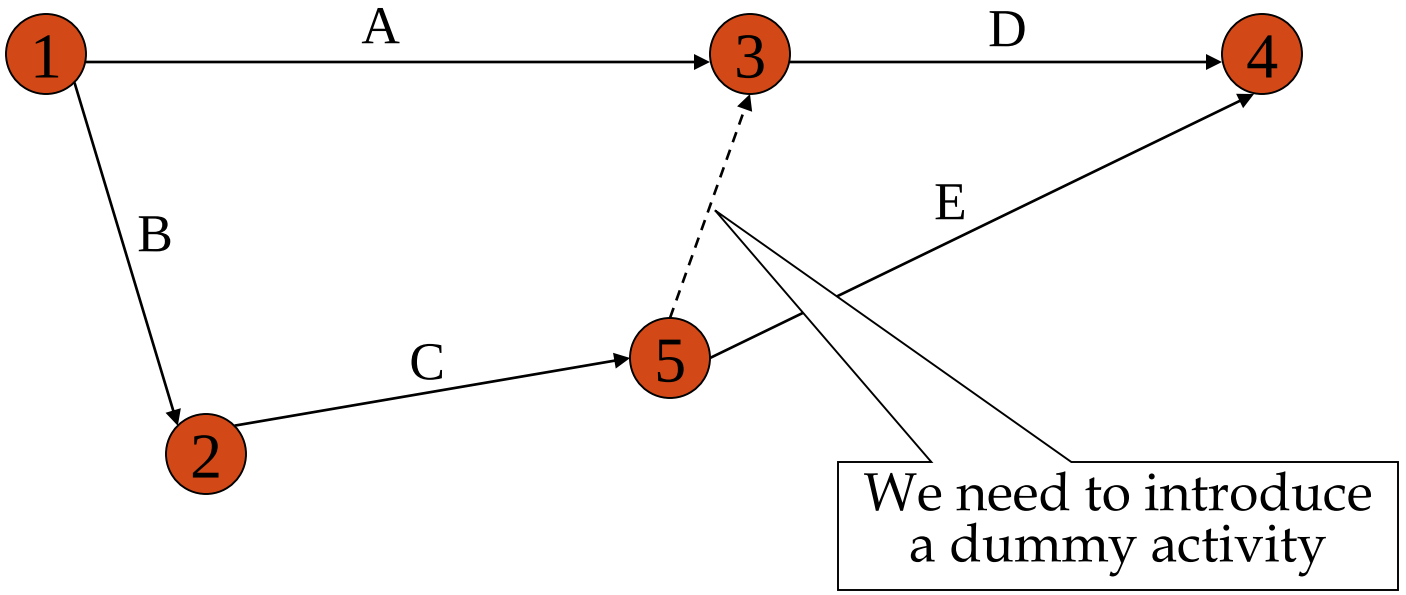
EXAMPLE 2

Develop the network for a project with following activities and immediate predecessors:

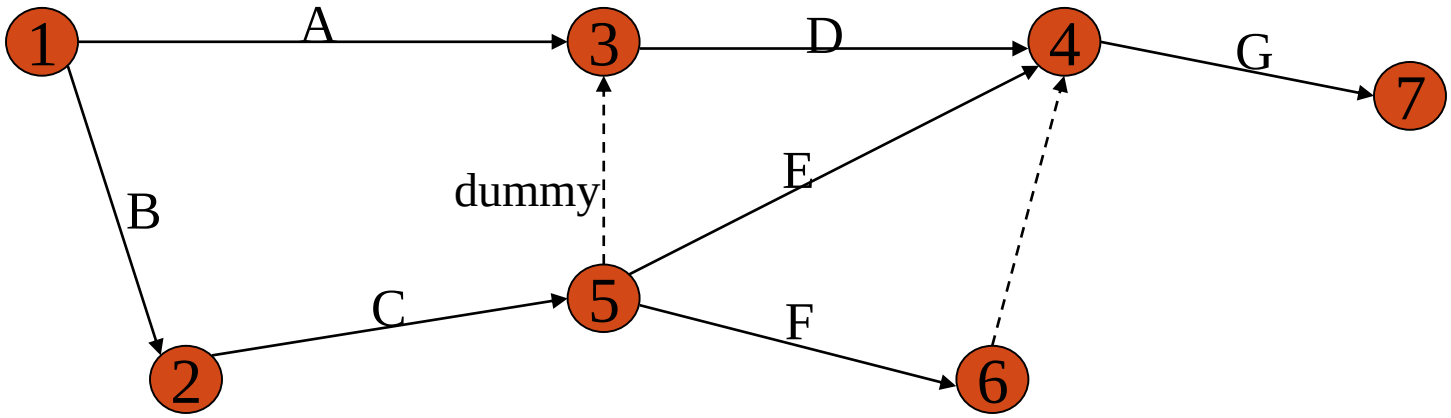
Activity	Immediate predecessors
A	-
B	-
C	B
D	A, C
E	C
F	C
G	D,E,F

Try to do for the first five (A,B,C,D,E) activities

NETWORK OF FIRST FIVE ACTIVITIES



Network of Seven Activities



- Note how the network correctly identifies D, E, and F as the immediate predecessors for activity G.

- Dummy activities is used to identify precedence relationships correctly and to eliminate possible confusion of two or more activities having the same starting and ending nodes

- Dummy activities have no resources (time, labor, machinery, etc) – purpose is to PRESERVE LOGIC of the network

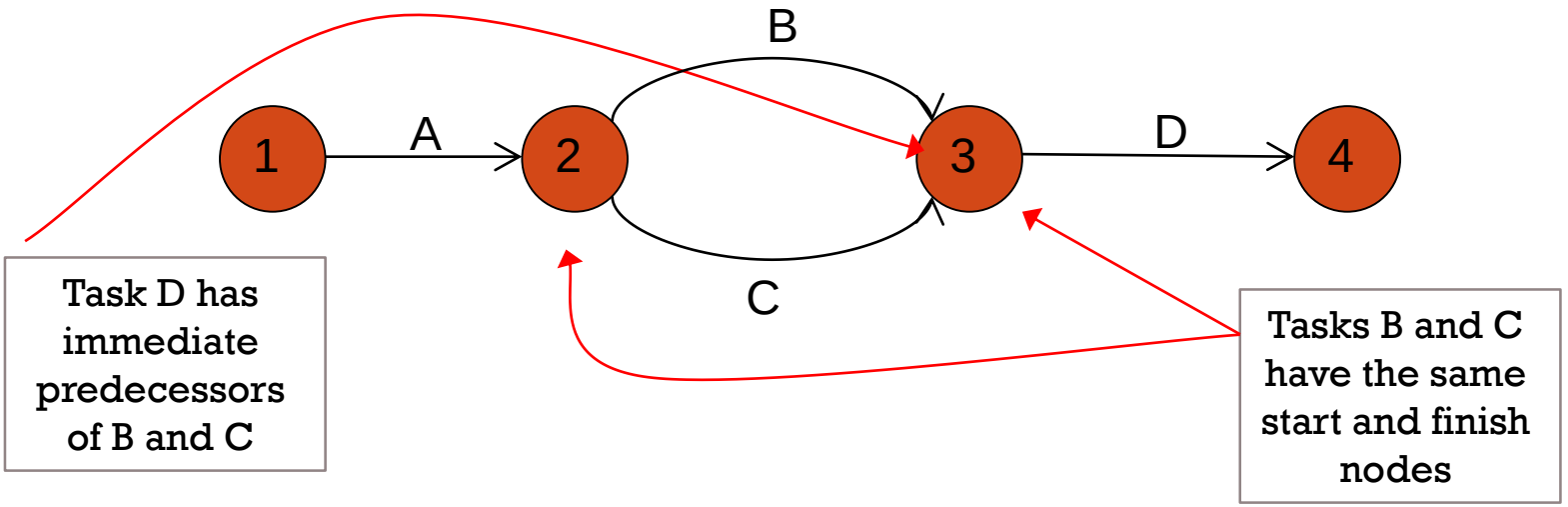
DUMMY ACTIVITIES

- Sometimes it is necessary to insert dummy activities (duration zero) in order to maintain the clarity of the diagram and the precedence relationships between activities.
- In activity-on-arrow PERT diagrams, each activity must be uniquely identifiable by its start and end nodes.
- However, sometimes multiple tasks have the same predecessors and successors.

DUMMY ACTIVITIES

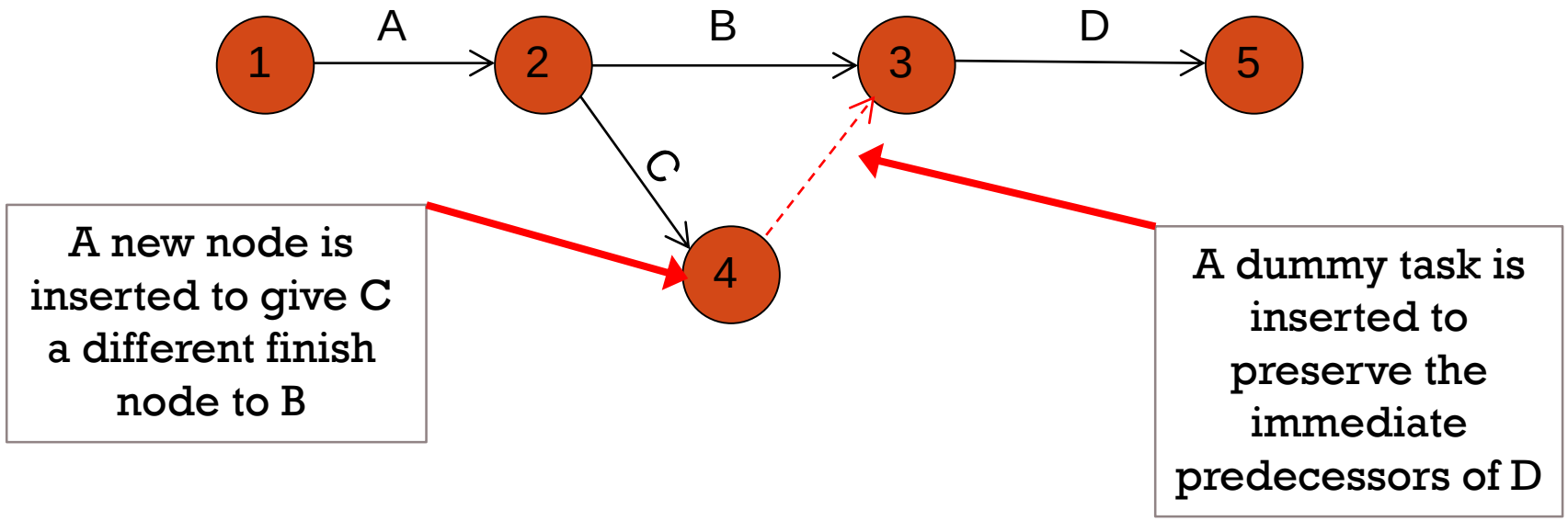
Case One

- A task should be uniquely identifiable from its start node and finish node
  - ✓ *This means that two or more tasks cannot share the same start and finish nodes*



DUMMY ACTIVITIES

- Inserting a *dummy activity* can ensure that multiple tasks have different successors.

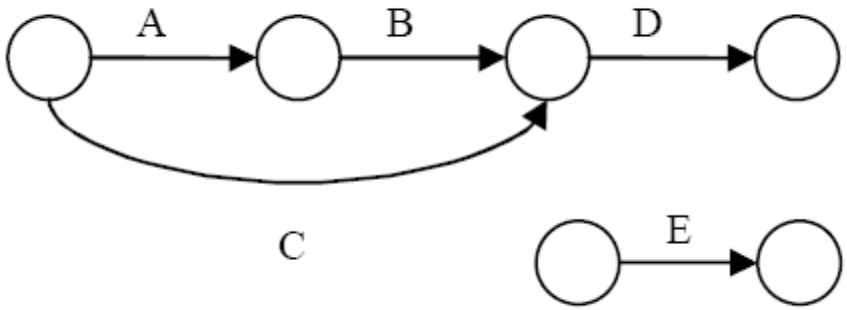


DUMMY ACTIVITIES

Case Two

- A task should be uniquely identifiable from its start node and finish node
  - *This also means that a task cannot have more than one start node and one finish node*
  - *In other words… two different arrows cannot represent the same task.*

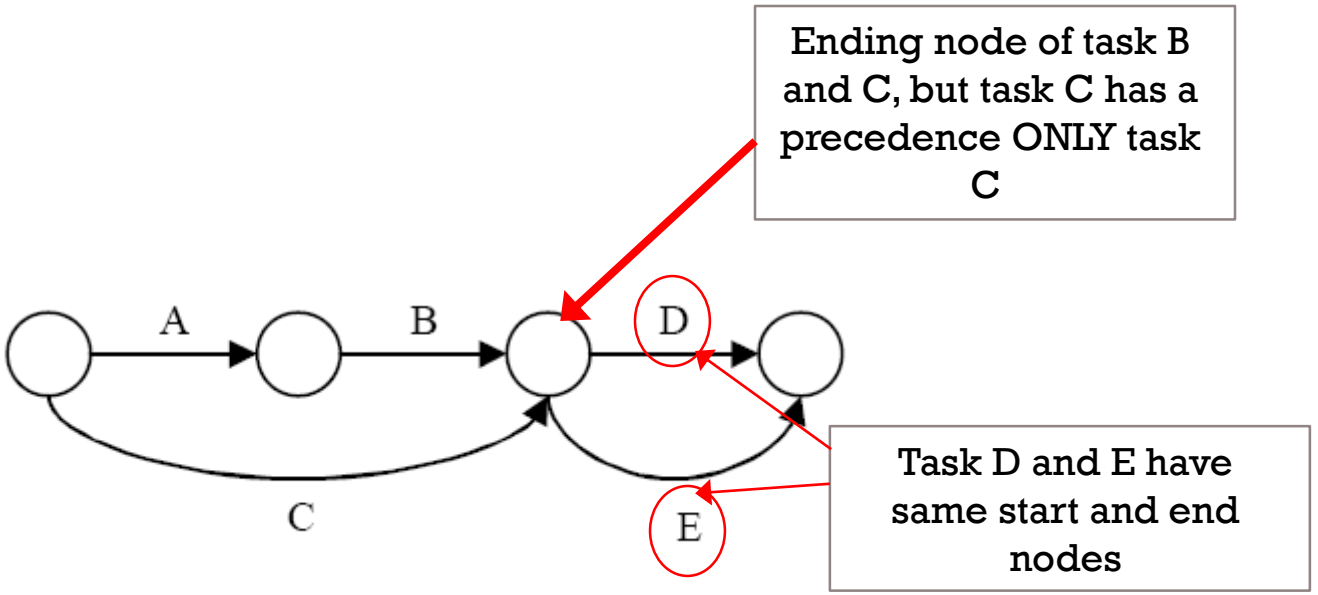
Task	Precedence
A	
B	A
C	
D	B,C
E	C



DUMMY ACTIVITIES

Possible Solution One

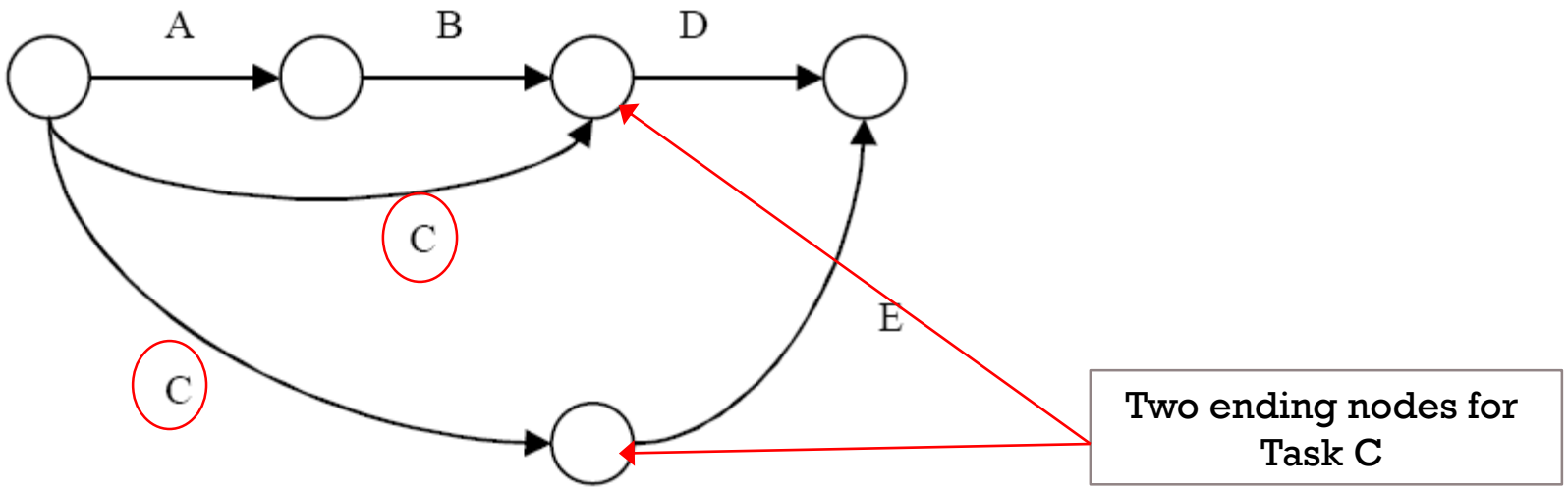
- One option would be to insert activity E as shown below, but this changes the precedence of E and provokes that D and E share both start and end nodes.



DUMMY ACTIVITIES

Possible Solution Two

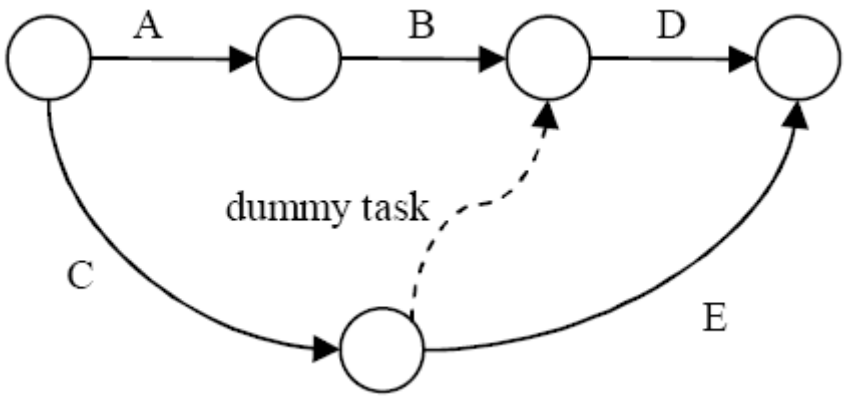
- Another option would be to insert activity E as shown below, but this provokes C to have two different end nodes.



DUMMY ACTIVITIES

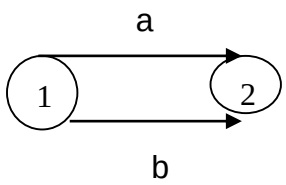
Correct Solution!

- The solution is to insert a dummy task so that the precedence of E is preserved and activity C remains uniquely identifiable.

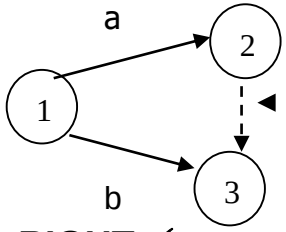


EXAMPLES OF THE USE OF DUMMYACTIVITY

Network concurrent activities



WRONG!!!



Dummy

RIGHT ✓

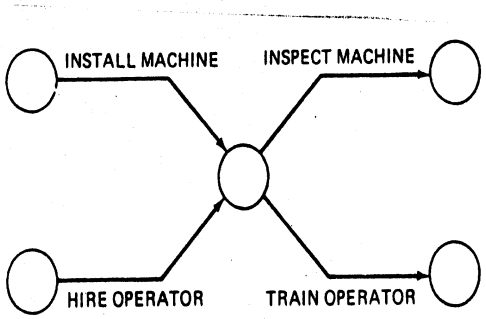
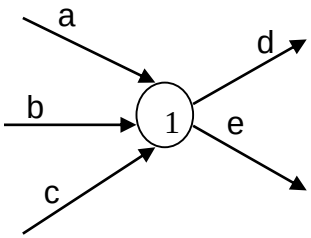


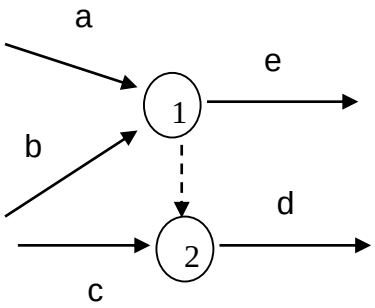
Figure 2-16

WRONG !

Activity c not required for e



WRONG !!!



RIGHT ✓

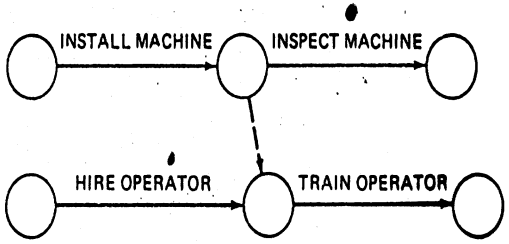
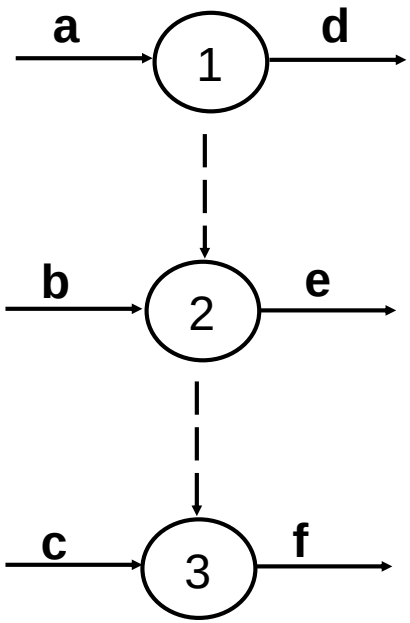


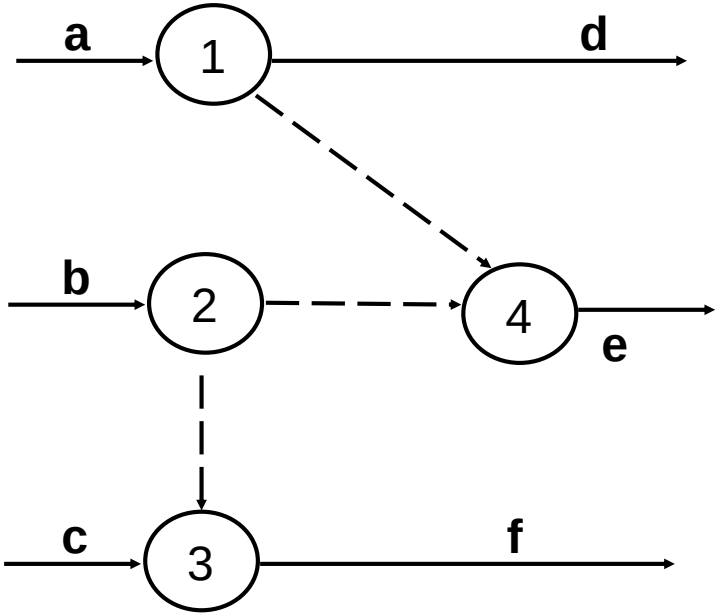
Figure 2-17

RIGHT ✓

WRONG!!!



RIGHT!!!



a precedes **d.**
a and **b** precede **e,**
b and **c** precede **f** (**a** does not precede **f**)

SCHEDULING WITH ACTIVITY TIME

Activity	Immediate predecessors	Completion Time (week)
A	-	5
B	-	6
C	A	4
D	A	3
E	A	1
F	E	4
G	D,F	14
H	B,C	12
I	G,H	2
	Total ……	51

This information indicates that the total time required to complete activities is 51 weeks. However, we can see from the network that several of the activities can be conducted simultaneously (A and B, for example).

EARLIEST START & EARLIEST FINISH TIME

- We are interested in the longest path through the network, i.e., the critical path.

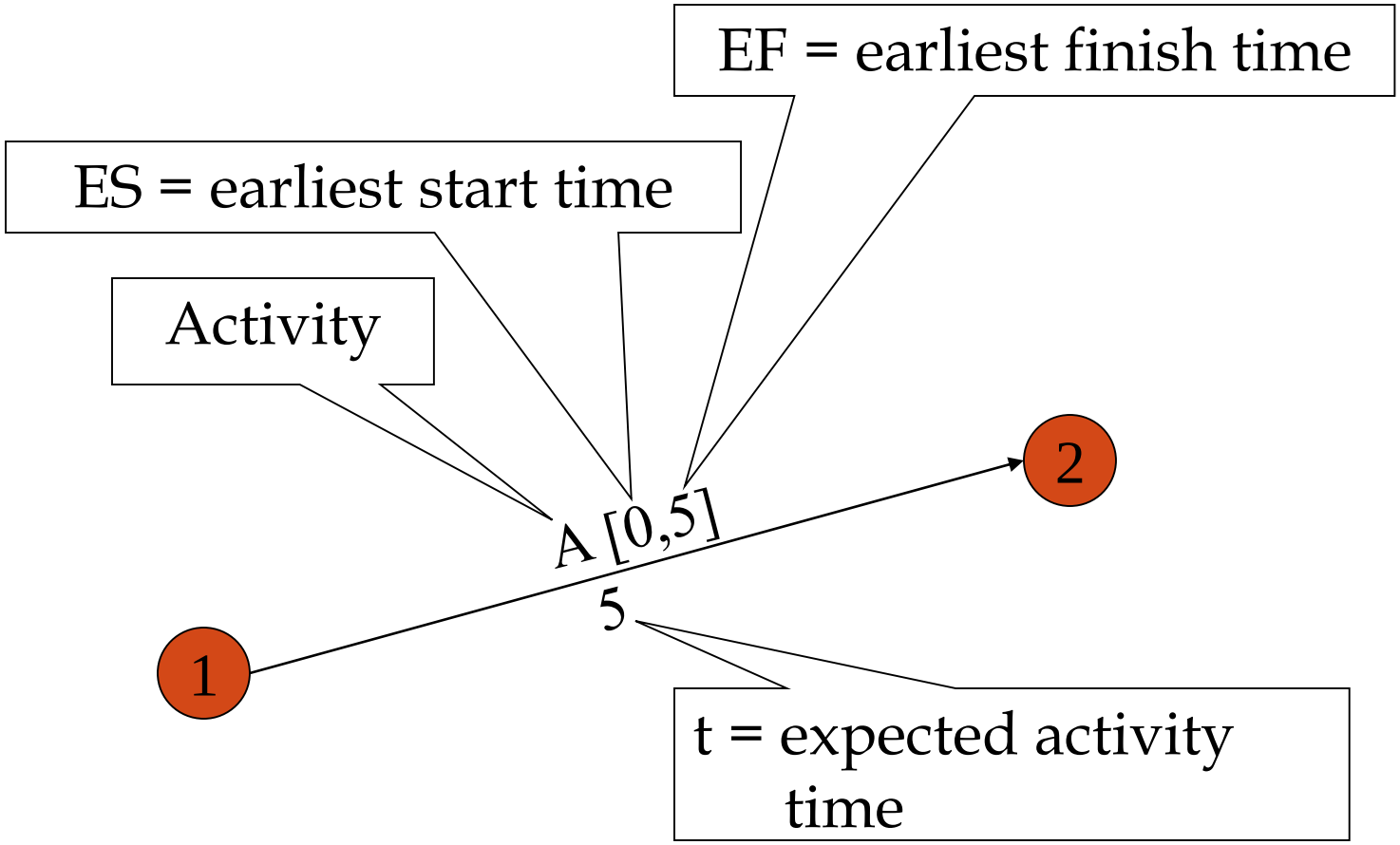
- Starting at the network's origin (node 1) and using a starting time of 0, we compute an ***earliest start*** (ES) and ***earliest finish*** (EF) time for each activity in the network.

- The expression EF = ES + t can be used to find the earliest finish time for a given activity.

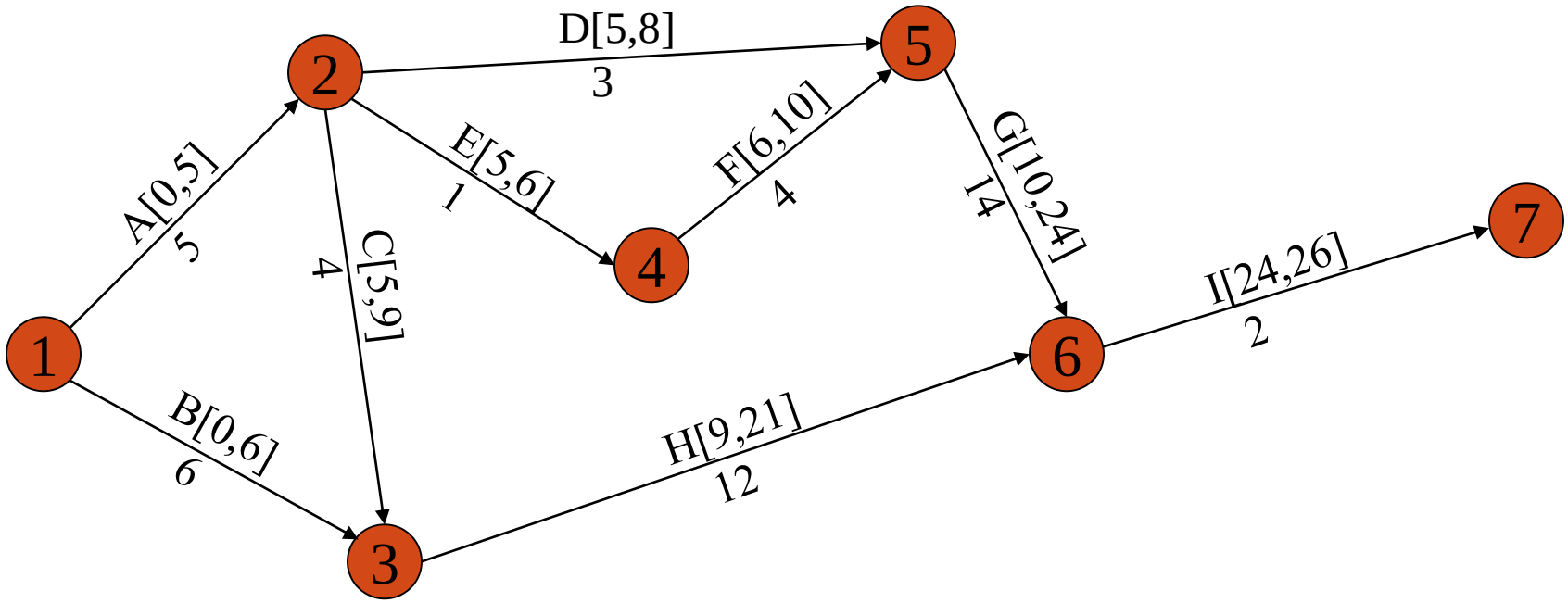
  For example, for activity A, ES = 0 and t = 5; thus the earliest finish time for activity A is

$$EF = 0 + 5 = 5$$

ARC WITH ES & EF TIME



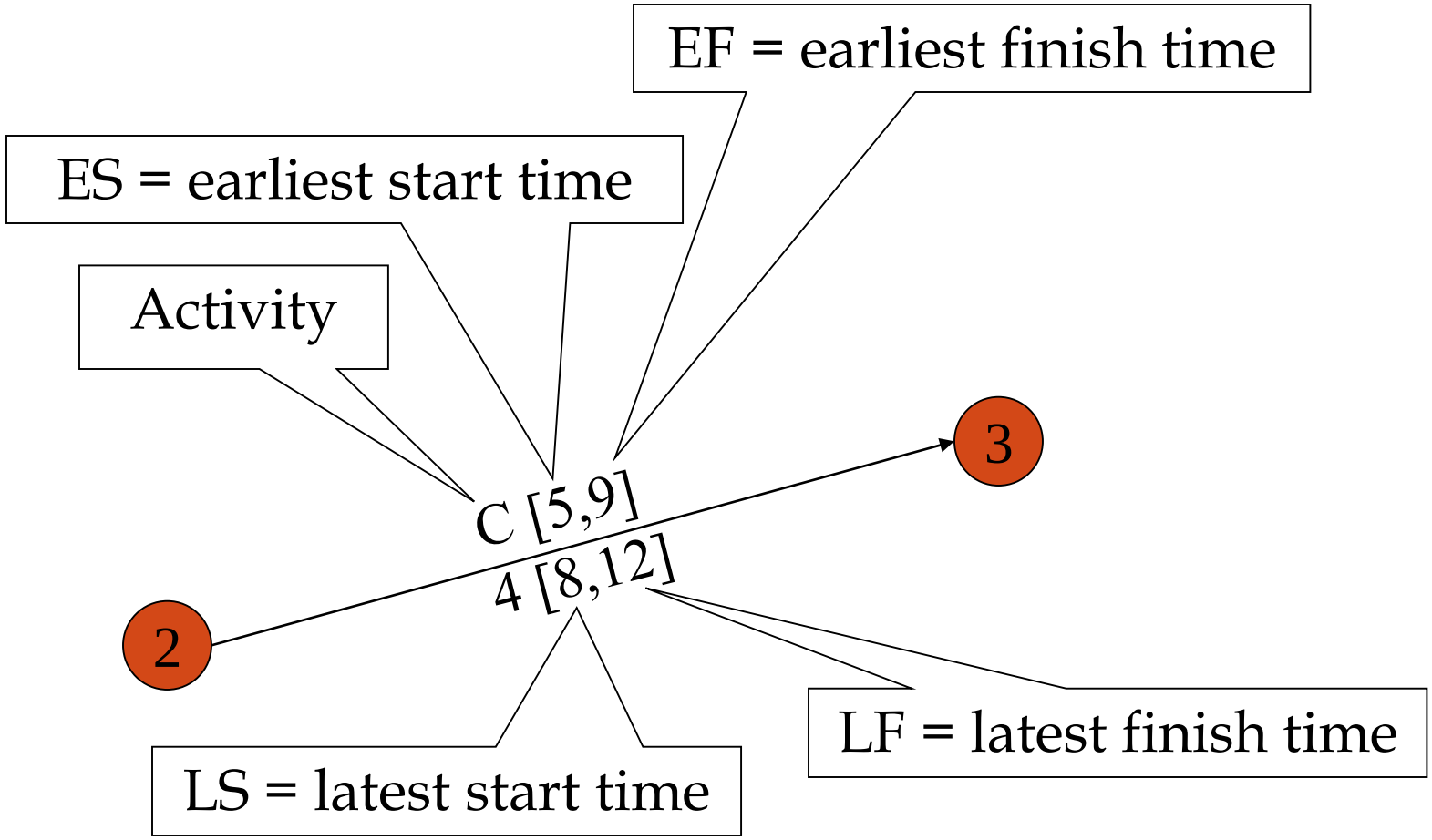
NETWORK WITH ES & EF TIME



Earliest start time rule:

The earliest start time for an activity leaving a particular node is equal to the **largest** of the earliest finish times for all activities entering the node.

ACTIVITY, DURATION, ES, EF, LS, LF



LATEST START & LATEST FINISH TIME

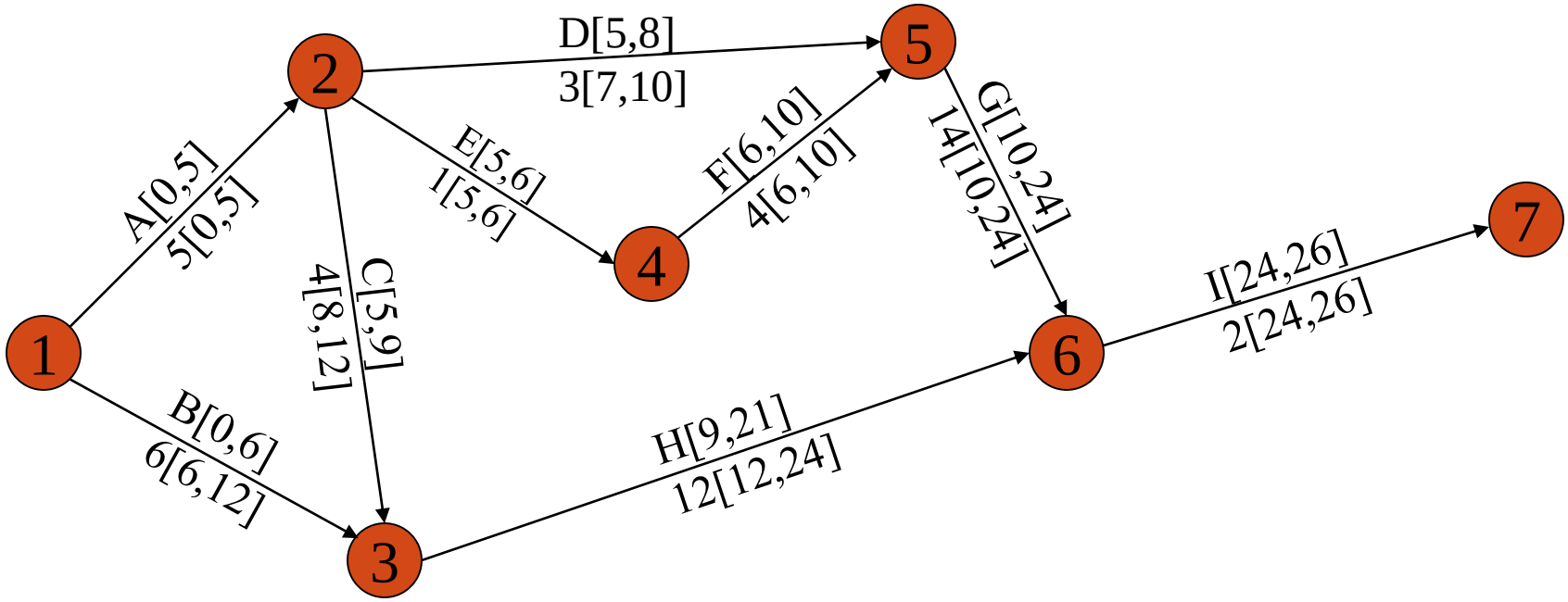
- To find the critical path we need a backward pass calculation.

- Starting at the completion point (node 7) and using a ***latest finish*** time (LF) of 26 for activity I, we trace back through the network computing a ***latest start*** (LS) and latest finish time for each activity

- The expression **LS = LF – t** can be used to calculate latest start time for each activity. For example, for activity I, LF = 26 and t = 2, thus the latest start time for activity I is

$$LS = 26 - 2 = 24$$

NETWORK WITH LS & LF TIME



Latest finish time rule:

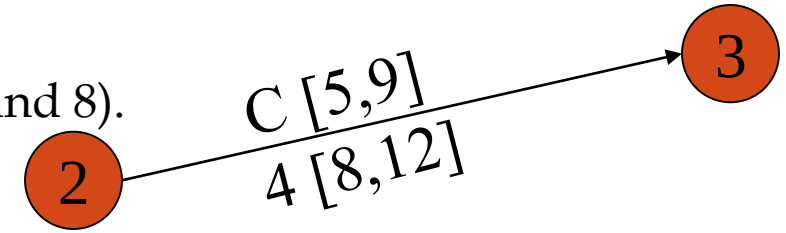
The latest finish time for an activity entering a particular node is equal to the **smallest** of the latest start times for all activities leaving the node.

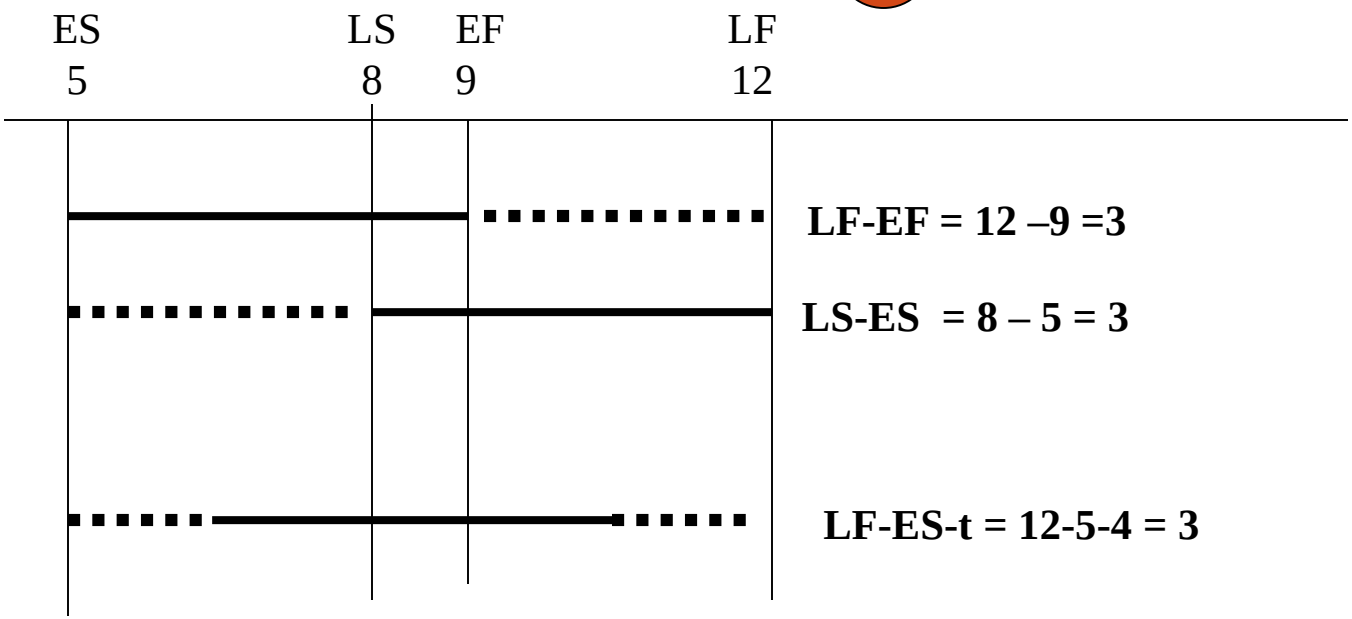
SLACK OR FREE TIME OR FLOAT

Slack is the length of time an activity can be delayed without affecting the completion date for the entire project.

For example, slack for C = 3 weeks, i.e Activity C can be delayed up to 3 weeks

(start anywhere between weeks 5 and 8).





ACTIVITY SCHEDULE FOR OUR EXAMPLE

Activity	Earliest start (ES)	Latest start (LS)	Earliest finish (EF)	Latest finish (LF)	Slack (LS-ES)	Critical path
A	0	0	5	5	0	Yes
B	0	6	6	12	6	
C	5	8	9	12	3	
D	5	7	8	10	2	
E	5	5	6	6	0	Yes
F	6	6	10	10	0	Yes
G	10	10	24	24	0	Yes
H	9	12	21	24	3	
I	24	24	26	26	0	Yes

IMPORTANT QUESTIONS

- **What is the total time to complete the project?**
  - 26 weeks if the individual activities are completed on schedule.

- **What are the scheduled start and completion times for each activity?**
  - ES, EF, LS, LF are given for each activity.

- **What activities are *critical* and must be completed as scheduled in order to keep the project on time?**
  - Critical path activities: A, E, F, G, and I.

- **How long can *non-critical* activities be delayed before they cause a delay in the project's completion time**
  - Slack time available for all activities are given.

IMPORTANCE OF FLOAT (SLACK) AND CRITICAL PATH

1. Slack or Float shows how much allowance each activity has, i.e how long it can be delayed without affecting completion date of project

2. Critical path is a sequence of activities from start to finish with zero slack. Critical activities are activities on the critical path.

3. Critical path identifies the minimum time to complete project

4. If any activity on the critical path is shortened or extended, project time will be shortened or extended accordingly

IMPORTANCE OF FLOAT (SLACK) AND CRITICAL PATH (CONT)

5. So, a lot of effort should be put in trying to control activities along this path, so that project can meet due date. If any activity is lengthened, be aware that project will not meet deadline and some action needs to be taken.

6. If can spend resources to speed up some activity, do so only for critical activities.

7. Don't waste resources on non-critical activity, it will not shorten the project time.

8. If resources can be saved by lengthening some activities, do so for non-critical activities, up to limit of float.

9. Total Float belongs to the path

CRITICAL PATH ANALYSIS (PERT)

Activity	LS	ES	Slacks	Critical ?
a	**0**	**0**	**0**	**Yes**
b	1	0	1	
c	4	0	4	
d	**20**	**20**	**0**	**Yes**
e	25	20	5	
f	29	20	9	
g	21	20	1	
h	14	10	4	
i	25	24	1	
j	**35**	**35**	**0**	**Yes**

COMPARISON BETWEEN CPM AND PERT

	CPM	PERT
1	Uses network, calculate float or slack, identify critical path and activities, guides to monitor and controlling project	Same as CPM
2	Uses one value of activity time	Requires 3 estimates of activity time Calculates mean and variance of time
3	Used where times can be estimated with confidence, familiar activities	Used where times cannot be estimated with confidence. Unfamiliar or new activities
4	Minimizing cost is more important	Meeting time target or estimating percent completion is more important
5	Example: construction projects, building one off machines, ships, etc	Example: Involving new activities or products, research and development etc

BENEFITS OF CPM / PERT NETWORK

Consistent framework for planning, scheduling, monitoring, and controlling project.

- Shows interdependence of all tasks, work packages, and work units.
- Helps proper communications between departments and functions.
- Determines expected project completion date.
- Identifies so-called critical activities, which can delay the project completion time.

BENEFITS OF CPM / PERT NETWORK (CONT.)

- Identified activities with slacks that can be delayed for specified periods without penalty, or from which resources may be temporarily borrowed
- Determines the dates on which tasks may be started or must be started if the project is to stay in schedule.
- Shows which tasks must be coordinated to avoid resource or timing conflicts.
- Shows which tasks may run in parallel to meet project completion date