

Schedule the Project

A project schedule requires 3 pieces of information:

- > A list of project activities
- > Activity duration
- > Activity relationships (dependencies)

Scheduling Methods

- There are two basic methods:
 - > The bar chart method
 - > The CPM methods based on networks:
 - The Arrow Network Diagramming method
 - The Precedence Network Diagramming method

The Bar Chart Method

- The bar chart method portrays a project activity by a bar extend over the duration of the activity.
- It has two major advantages
 - > It is simple to make as well as to read.
 - > It provides a clear visual indication of the project

The Bar Chart Method

- The bar chart has the following disadvantages:
 - > The relationship between activities is not shown. Therefor the effect of delay of an activity on the project duration or on another activity cannot be determined. Also, the impact of a change on project duration cannot be known.
 - > Does not show sufficient detail to enable timely detection of schedule slippage.
 - > Good for small projects. Large or complex projects tend to produce an unwieldy bar chart schedules reducing their visual clarity and simplicity.
- Bar charts are widely used now in conjunction with the CPM methods. In this case the bar chart is constructed using the schedule dates obtained from the CPM schedule. Therefore the disadvantages are mitigated.

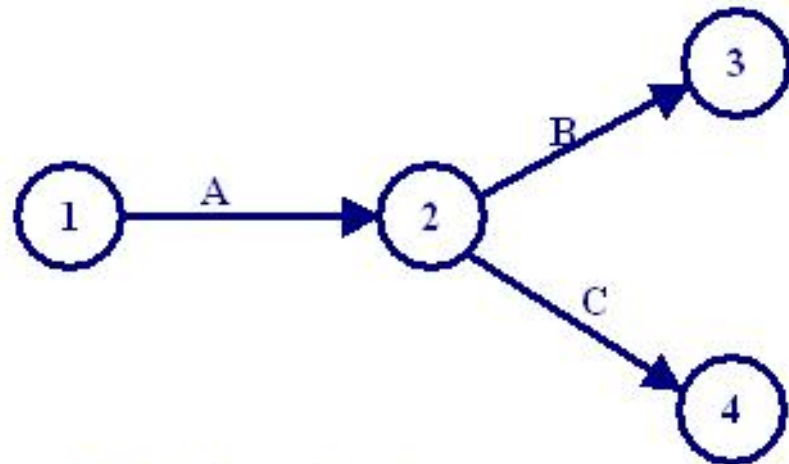
Arrow Diagram Networks



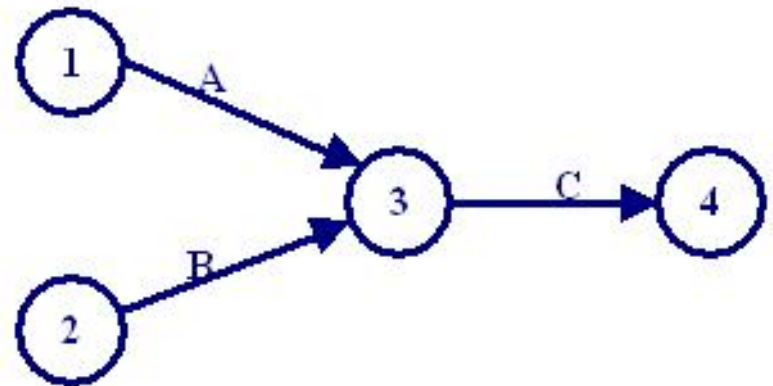
An activity representation in an ADN



B depends on A

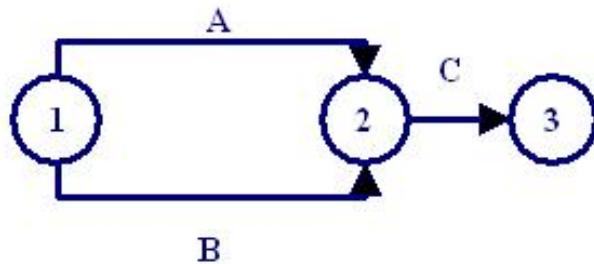


B & C depend on A

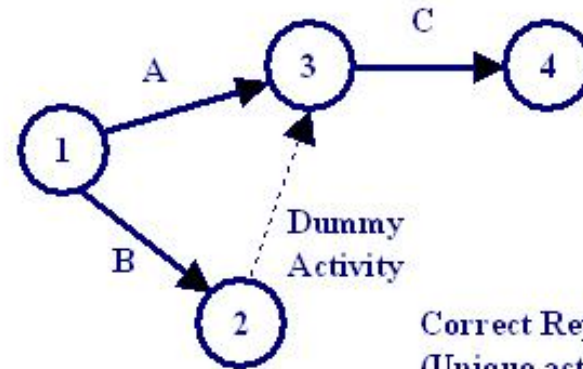


C depend on C & A

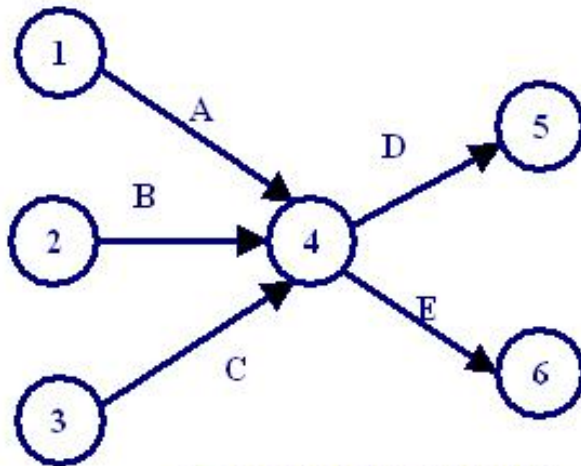
Arrow Diagrams Basic Patterns



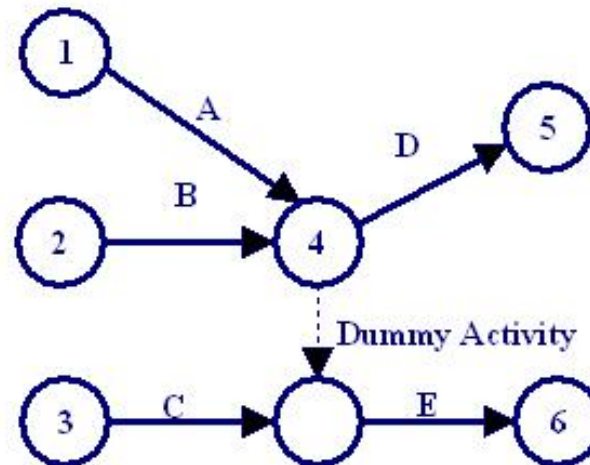
A and B share the same 2 nodes
This is not allowed



Correct Representation
(Unique activity Numbers)



"D does not depend on C"
Incorrect Representation



Correct Diagram
(Correct Logic)

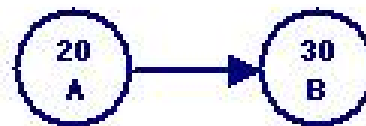
Rules for Arrow Diagram Networks

- Before an activity may begin, *ALL* activities preceding it must be completed. Activities, which have no predecessor, begin at the start of the project.
- The length of the arrow or its direction has no significance.
- Event numbers must be unique.
- No two events (nodes) can be connected by more than one activity.
- A network must start by only *one initial event* and must end in only *one end event*.
- Numbers should be numbered so that the node at the head has a higher number than the node at the tail.
- Dummies should be used only when necessary. Redundant dummies should be removed. Remember dummies are used for one of two reasons only: Maintain *unique numbering* and define *correct logic*.

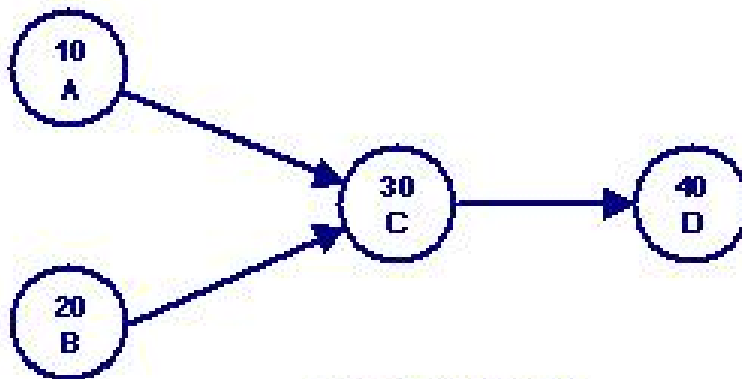
Precedence Diagram Networks

- In precedence diagrams the *activity* is represented by the *node* and the *link* represent the *relationships* between activities.
- The basic activity representation and logic patterns are as shown in the diagrams below.

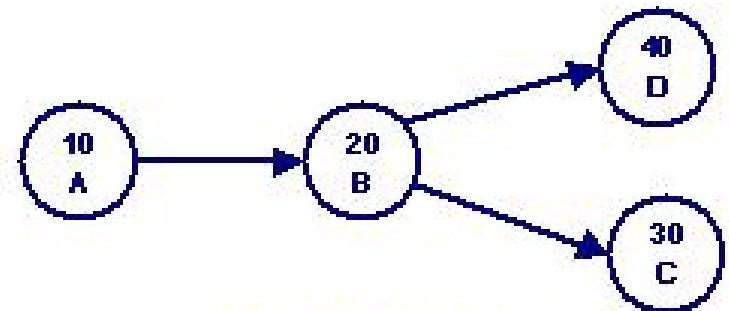
The Precedence Diagrams



Activity B depends on Activity A



A merge relationship

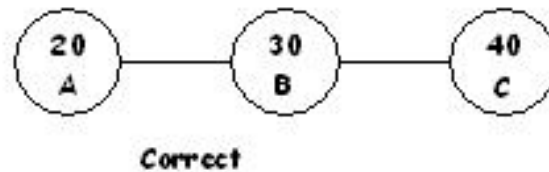
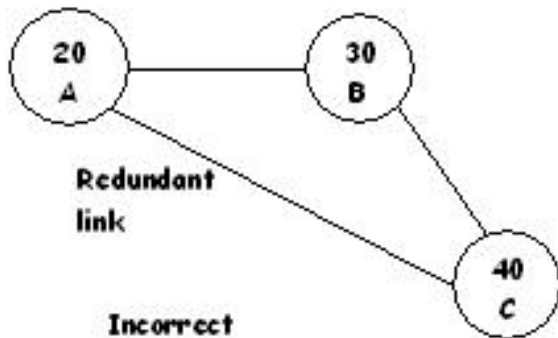


A burst relationship

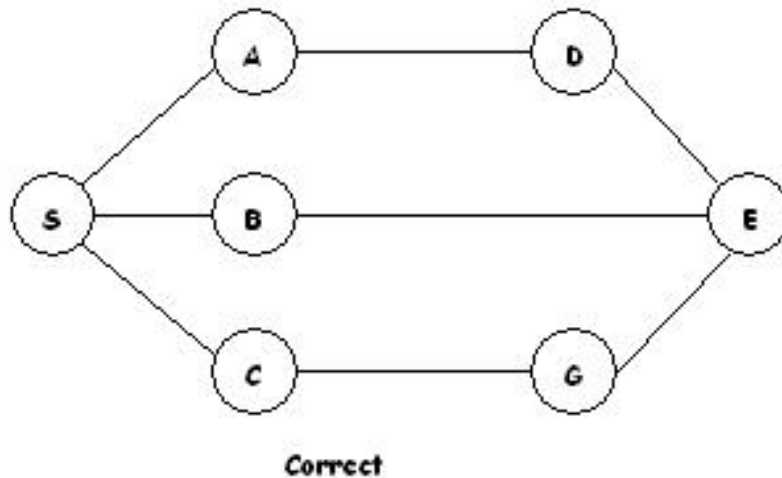
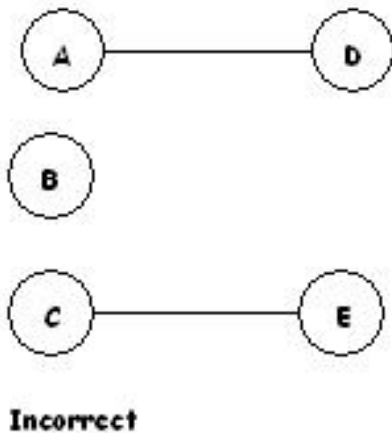
Rules for constructing precedence diagrams

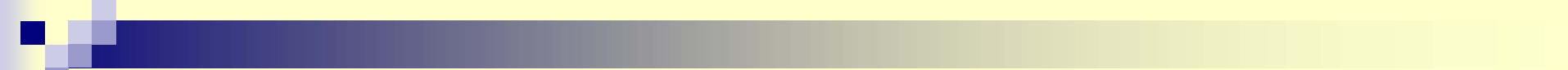
- Activity numbers must be unique.
- A network must start by one activity and ends by one activity.
- In a project where several activities have no predecessors a dummy (fictitious) activity is introduced with links to all the starting activities.
- Similarly, where several activities end the project a dummy activity is introduced as a terminal activity. The other activities are linked to it.
- There is no other use for a dummy activity other than the above two in a precedence diagram.

1. Remove redundant

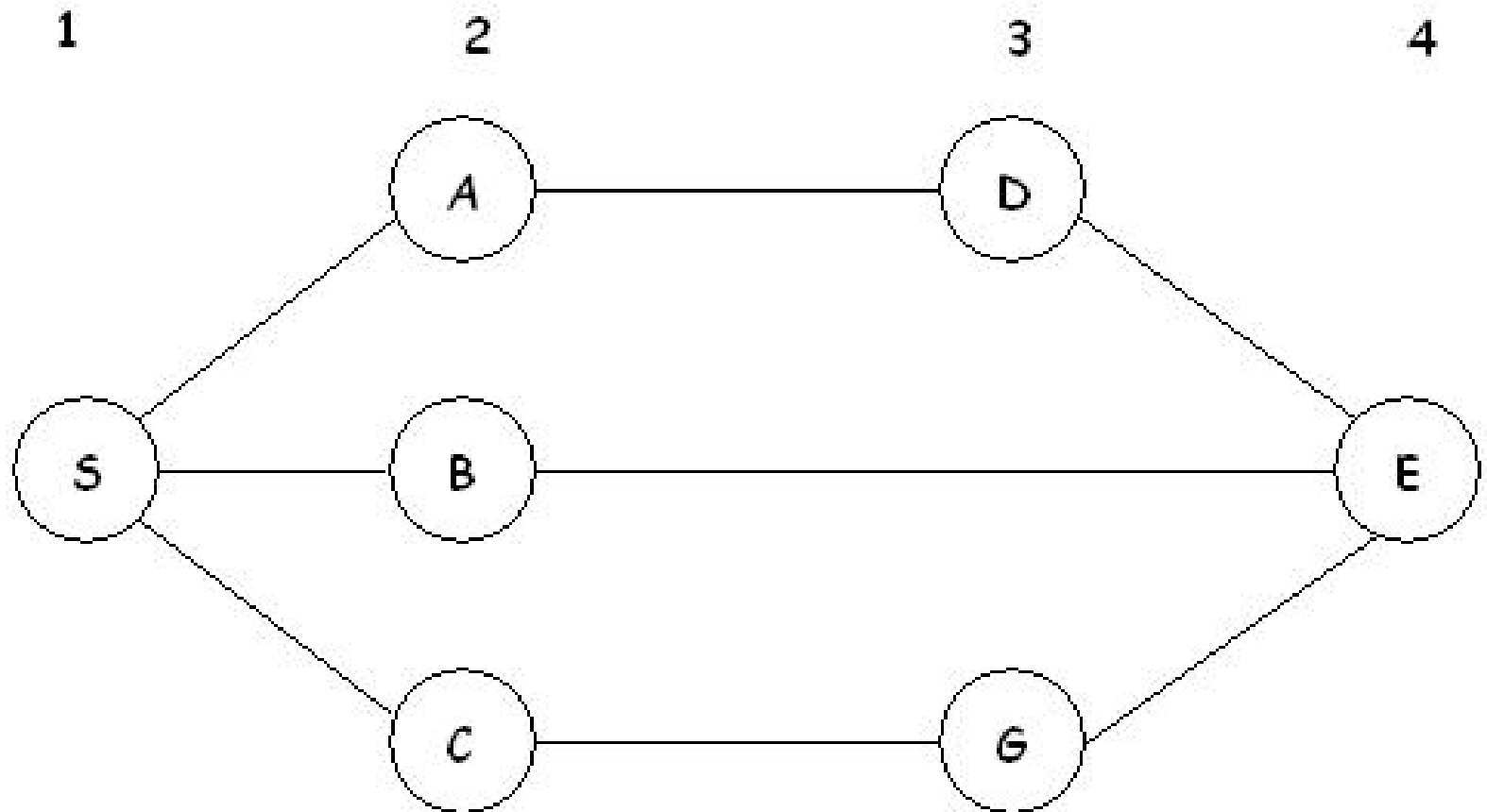


2. Start project with a single activity and end project by one



- 
- Activities in a precedence diagram are usually arranged in sequence steps (see diagram)
 - The sequence step of an activity is the maximum of the sequence steps of activities preceding it plus one.

Sequence Steps



Precedence diagram with activities arranged in sequence