Software Project Management (SPM)

Course Code: CACS407
Year/ Semester: IV/VII

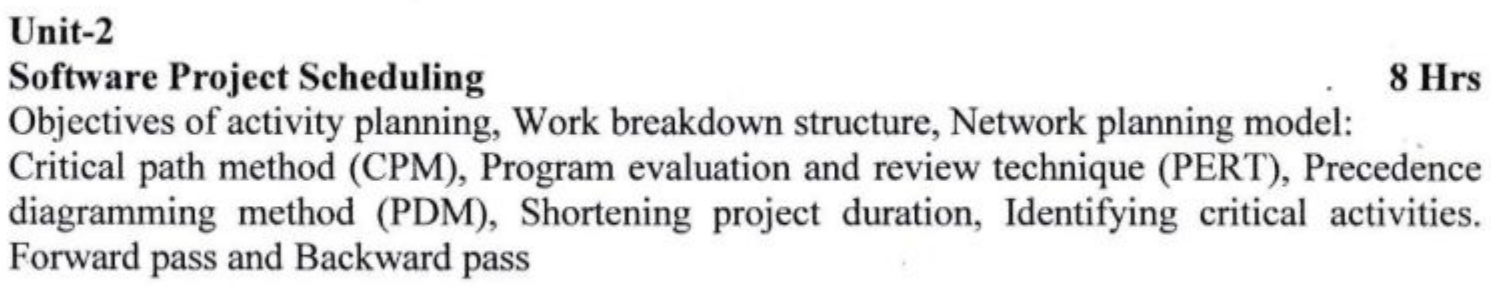
Compiled by Shishir Ghimire

Credit Hours: 3hrs

Unit - 02: Software Project Scheduling

Class Load : 8 hrs

TABLE OF CONTENTS



Unit-2

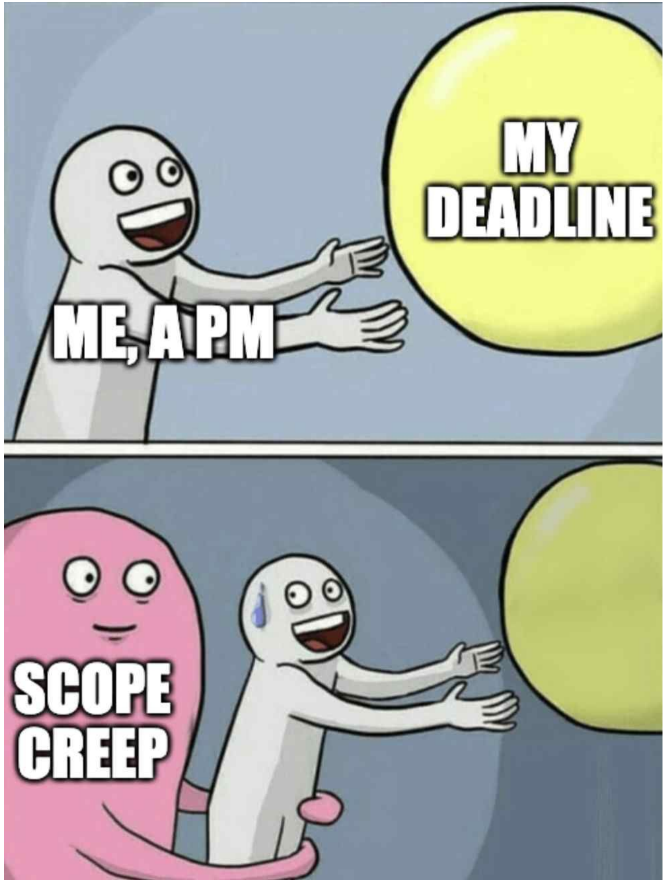
Software Project Scheduling **8 Hrs**

Objectives of activity planning, Work breakdown structure, Network planning model: Critical path method (CPM), Program evaluation and review technique (PERT), Precedence diagramming method (PDM), Shortening project duration, Identifying critical activities. Forward pass and Backward pass

Introduction:

▶ WHY?

WHEN THE DEV TEAM SAYS THEY CAN'T DELIVER THE PROJECT ON TIME
OL' RELIABLE
BONUSES
imgflip.com


ME, A PM
MY DEADLINE
SCOPE CREEP

Scheduling:

"I love deadlines. I love the whooshing sound they make as they fly by." – Douglas Adams

- ❖ The schedule connects the scope, work estimates and deadline into a network of software development tasks.
- ❖ Must Manage:
  - ➢ **Parallelism** (tasks can be undertaken simultaneously)
  - ➢ **Dependency** (task has an effect on subsequent tasks)
- ❖ **90-90 Rule:** First 90% of a project is complete in 90% of the scheduled time. The other 10% is also completed in 90% of the time.

Project Scheduling:

- Software Project Scheduling is an activity that distributes **Estimated Effort** across the **Planned Project duration** by allocating the effort to specific Software Engineering Task.
- Software Project Scheduling is the process of determining **when** project activities will be performed and in **which sequence**.
- It involves creating a timeline that outlines that start and end dates for each task or activity within the project, as well as dependencies between the task.
- Project Scheduling establishes a "**Road Map**" for the Project Manager when combined with **Estimation Methods** and **Risk Analysis**.

Key Aspects of Project Scheduling:

- **Define all Tasks**.
- **Build a Network** that depicts that's dependencies.
- **Identify the tasks** that are Critical within the Network.
- **Track task progress** to ensure that delay is recognized. "One Day at a time ".

- Task Sequencing
- Estimating Task Durations
- Resource Allocation
- Identifying Milestones
- Critical Path Analysis
- Contingency Planning
- Updating and Monitoring

Project Scheduling Techniques:

1. Program Evaluation and Review Technique (PERT)
2. Gantt Chart
3. Critical Path Method (CPM)
4. Work Breakdown Structure (WBS)
5. Network Diagram
6. Precedence Diagramming Method (PDM)

How do you track task?

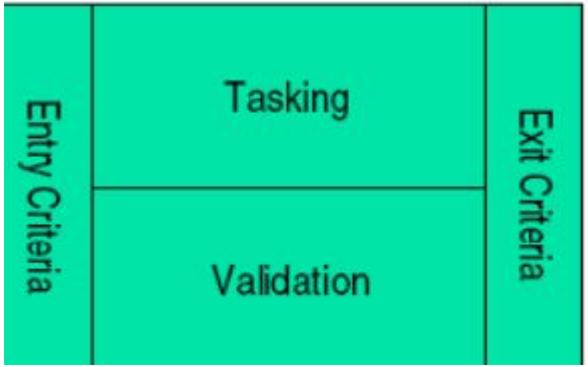
=> **ETVX**

E = **E**ntry Criteria (Before Starting)

T = **T**asking

V = **V**alidation

X = **E**xit Criteria (After Finished)



Entry Criteria

Tasking

Validation

Exit Criteria

Objective of Activity Planning

2.1

Activity Planning in SPM:

An activity plan should provide a means of **evaluating the consequences of not meeting any of the activity target dates** and guidance as to **how the plan might most effectively be modified to bring the project back to target.**

❖ Activity Planning is a critical component of software project management, ensuring that the project is completed **within time and within budget** while meeting the specified requirement.

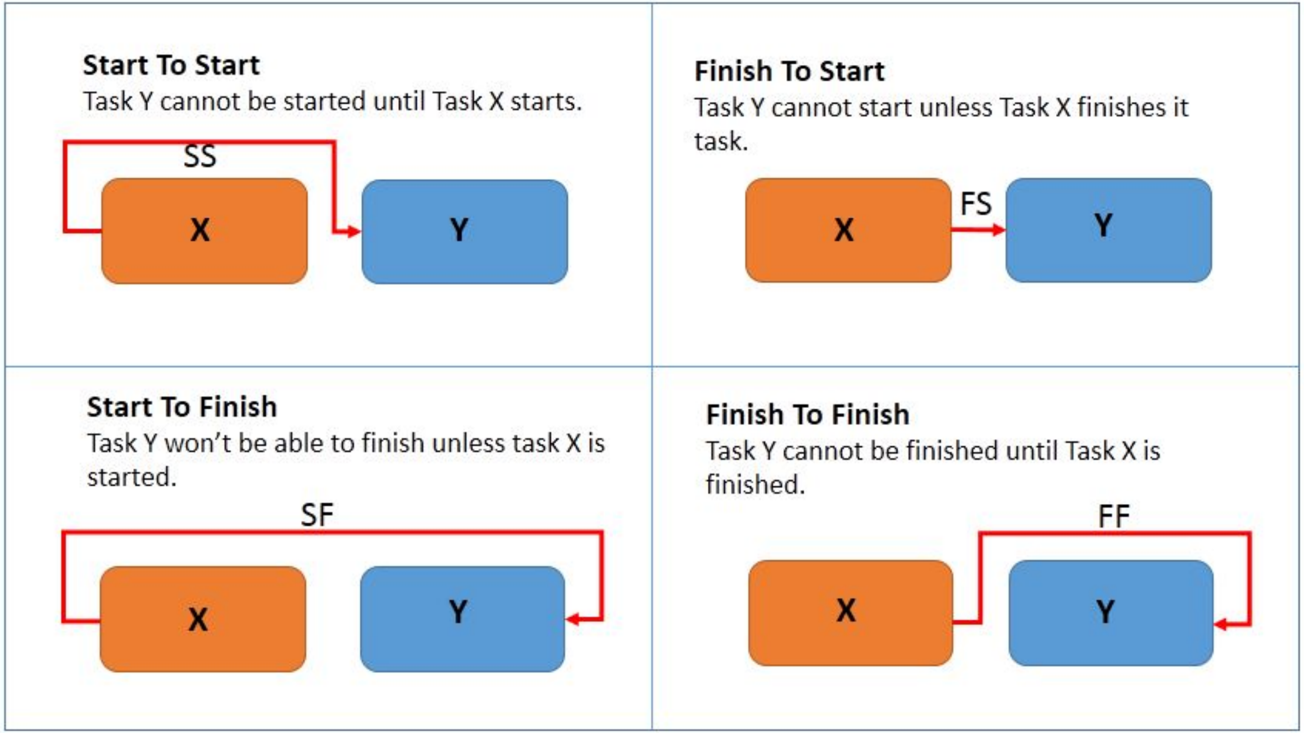
❖ Activity Planning breaks down into **two processes:**

- Defining Activities
- Sequencing Activities

Activity Planning in SPM:

- **Defining Activities:**
  - Activities are specific tasks required to achieve deliverables.
  - Begins with a **Work Breakdown Structure (WBS)** to identify required activities.
  - Uses progressive elaboration to refine plans as **new knowledge** emerges.
  - **Rolling Wave Planning** involves ongoing updates to ensure flexibility as the project progresses.
- **Sequencing Activities:**
  - Establishes **logical relationships** among activities to maximize efficiency within project constraints.
  - Uses tools to **define activities, sequence them, and estimate resources/durations** to develop a project schedule.
  - Activities and milestones are connected logically (e.g., **finish-to-start** or **finish-to-finish** relationships).

Task Dependency Types:



Start To Start

Task Y cannot be started until Task X starts.

Finish To Start

Task Y cannot start unless Task X finishes it task.

Start To Finish

Task Y won't be able to finish unless task X is started.

Finish To Finish

Task Y cannot be finished until Task X is finished.

Objectives of Activity Planning:

- **Feasibility Assessment:** Determines if the project is achievable within required **timescales** and **resource constraints.**
- **Resource Allocation:** Identifies the most **efficient ways** to allocate resources and aligns resource availability with project timelines.
- **Detailed Costing:** Provides **accurate estimates** of project costs and **their timing** after creating activity plans and assigning resources.
- **Motivation:** Setting **targets** and **monitoring progress** motivates staff, especially when they contribute to target-setting.
- **Coordination:** Ensures **effective communication** and **collaboration** across teams, especially in large projects, by scheduling staff availability and transfers.

Activity Planning:

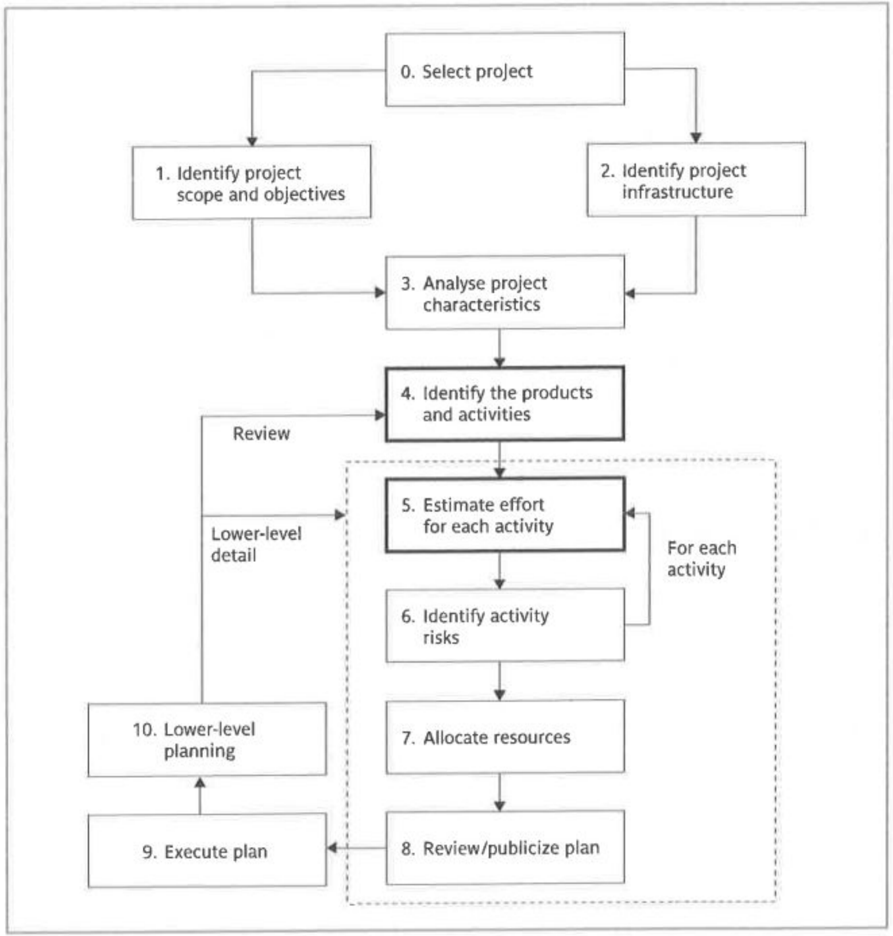

0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
10. Lower-level planning
7. Allocate resources
9. Execute plan
8. Review/publicize plan

FIGURE 6.1 Activity planning is carried out in Steps 4 and 5

When to Plan?

- ❖ Planning is an **ongoing process of refinement**, each iteration becoming more detailed and more accurate than the last.
- ❖ During feasibility study and project start up:
  - ➢ the main purpose of planning will be **to estimate timescales and the risks of not achieving target** completion dates or keeping within budget.
- ❖ As the project proceeds beyond the feasibility study, the emphasis will be placed upon the production of activity plans for **ensuring resource availability and cash flow control.**
- ❖ Throughout the project, until the final deliverable has reached the customer, **monitoring** and **re-planning** must continue to correct any **drift** that might prevent meeting time or cost targets.

Work Breakdown Structure (WBS)

2.2

Work Breakdown Structure (WBS):

A Work Breakdown Structure (WBS) is a fundamental tool in project management used to **break down large projects into smaller**, more manageable components.

- ❖ "Project Management Body of Knowledge **(PMBOK**)" is a group of terminology that describes the work breakdown structure as a "**deliverable-oriented hierarchical breakdown** of the work which is performed by the project team."
- ❖ A Work Breakdown Structure (WBS) is a hierarchical decomposition of a project into smaller, manageable components called **work packages**.
- ❖ Developed by the **U.S. Department of Defense (DoD**) in the late 1950s for use in the Polaris missile program.
- ❖ Adopted as a standard project management tool over time.

WBS Example 1:

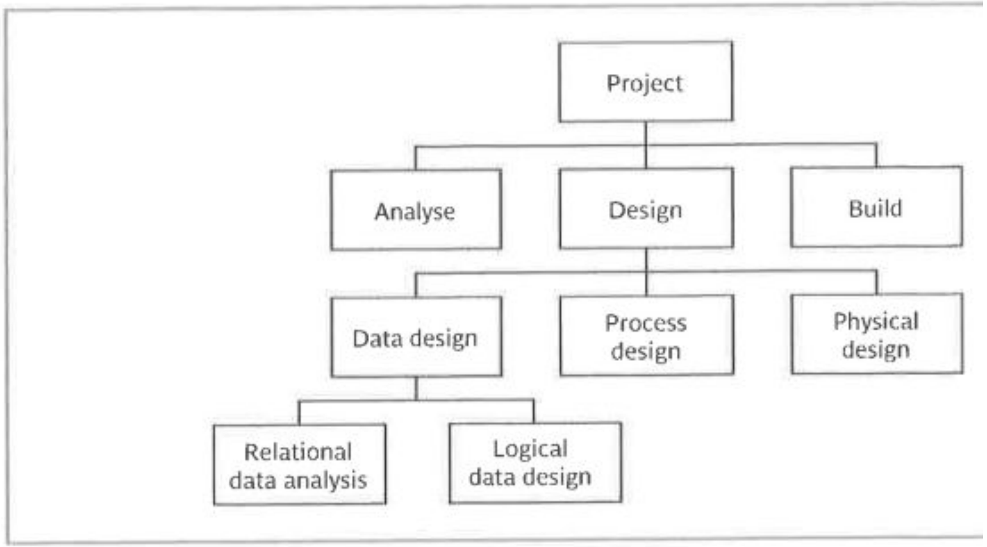
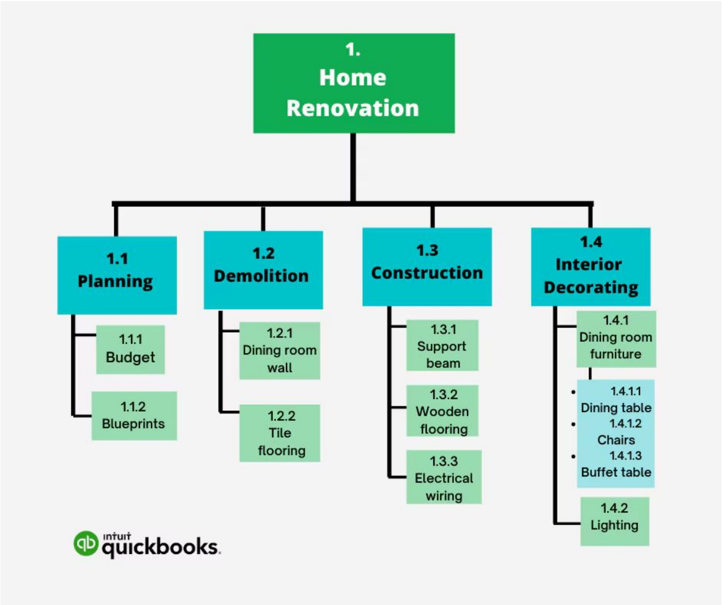
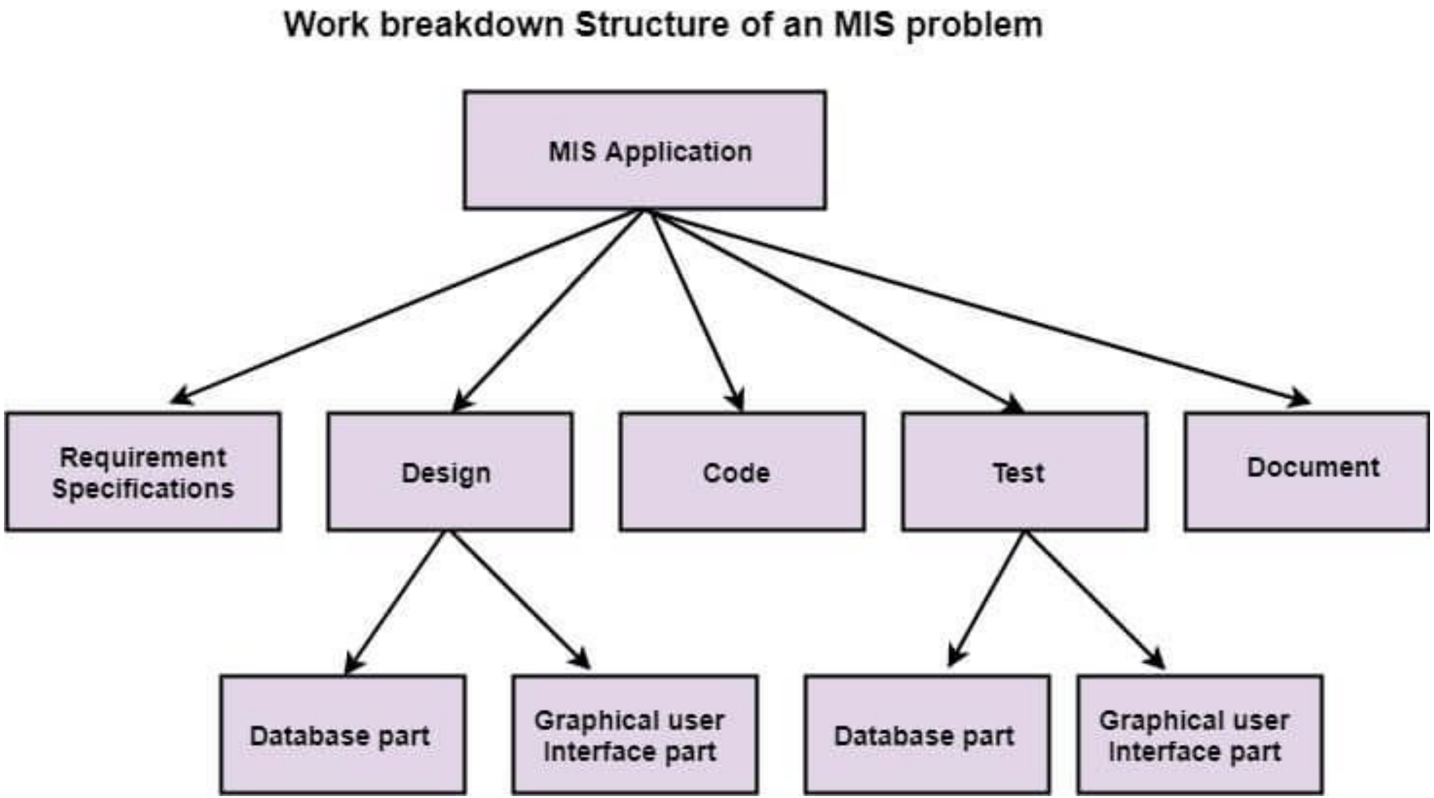


FIGURE 6.2 A fragment of an activity-based Work Breakdown Structure



WBS Example 2 :


Work breakdown Structure of an MIS problem
MIS Application
Requirement Specifications
Design
Code
Test
Document
Database part
Graphical user Interface part
Database part
Graphical user Interface part

Components of WBS :

- **Project Goal (Top-Level):**
  - Represents the **overall project** or objective to be achieved.
- **Phases or Deliverables:**
  - **Major sections** of the project that help in organizing work (e.g., Planning, Execution).
- **Work Packages:**
  - The **smallest units** of the WBS, representing tasks or activities that can be assigned and tracked. These are specific, measurable, and manageable.
- **Milestones:**
  - Key **checkpoints** or **events** that signify major achievements or progress in the project.
- **Dependencies:**
  - Relationships between tasks or activities to determine sequencing and timelines.
- **Unique Identifiers:**
  - Each component is often assigned a **unique code or ID** for tracking and reference.

Levels of WBS :

The WBS is hierarchical and typically consists of the following levels:

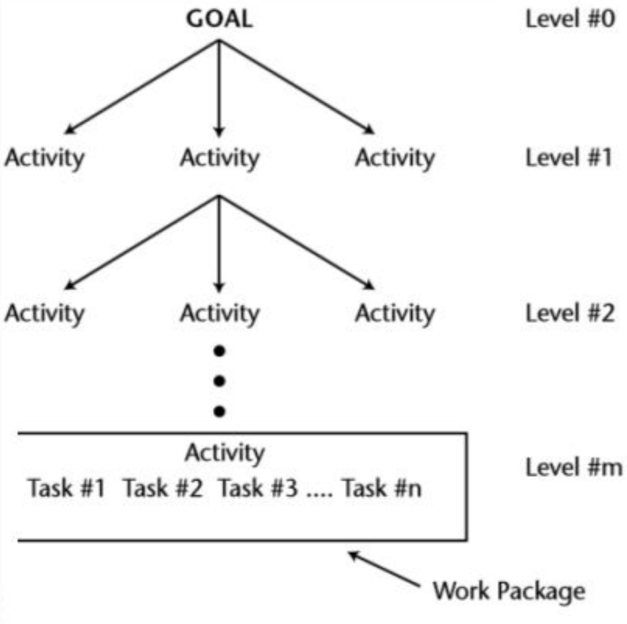
- **Level 1: Project Name or Objective**
  - Represents the **overall project**.
  - Example: "Website Development Project."
- **Level 2: Major Deliverables or Phases**
  - Breaks down the project into **broad categories** or **deliverables**.
  - Example: "Planning," "Design," "Development," "Testing."
- **Level 3: Sub-Deliverables**
  - Each phase or deliverable is further broken down into **smaller sub-components**.
  - Example: Under "Design," you may have "UI Design," "UX Design."
- **Level 4: Work Packages**
  - The **smallest components** of a WBS, representing activities or tasks.
  - Example: Under "UI Design," tasks might include "Create wireframes," "Develop prototypes."

Preparing a WBS :

In preparing a WBS there are a number of **steps** that need to be taken including:

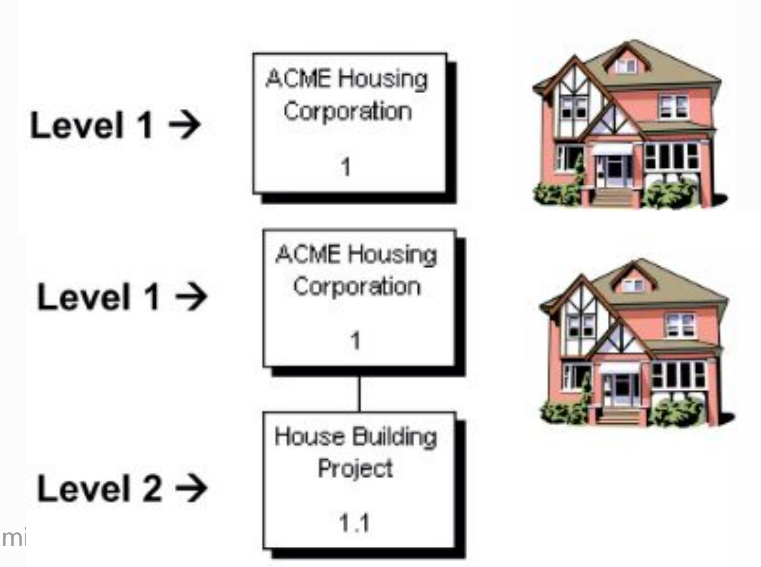
- **Identifying** the final project product
- **Divide** the total work of the project into major groups
- **Subdivide** these groups into tasks
- **Divide** these tasks into sub-tasks
- **Subtasks** should be small enough to permit adequate control and visibility

- **General Rule:** when an activity at Level n is decomposed into a set of activities at Level n + 1 and the work associated with those activities is complete, the activity at Level n, from which they were defined, is complete.



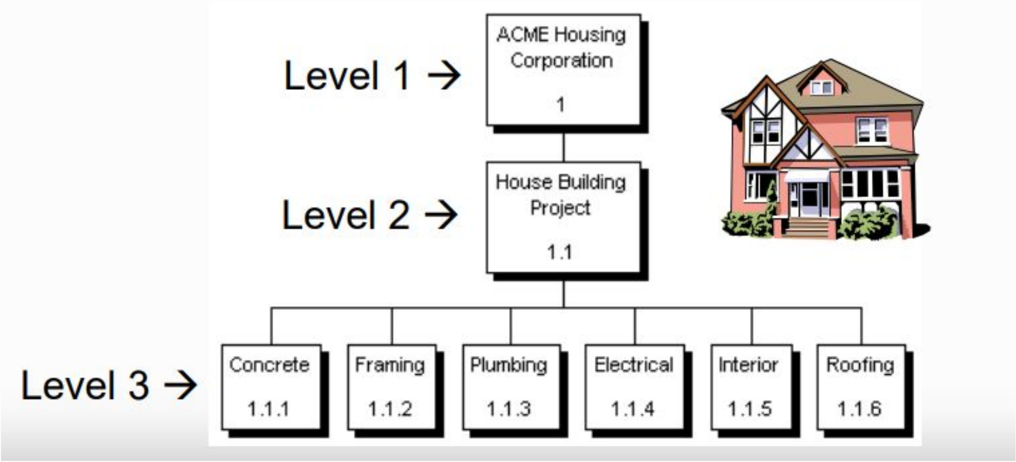
Building a WBS (Graphical Structure):

- Let's look at building a WBS for constructing a single family home.
- We hope that this is the first of many houses that ACME will build. So we start the WBS as **ACME** at the **highest position** or **Level 1** (It is given a code of 1 as all the future houses will be summarized at Level 1)
- **Level 2** is the level for each **individual project** (or house) that ACME undertakes. Level 2 always corresponds to Level 1. So code for Level 2 is 1.1

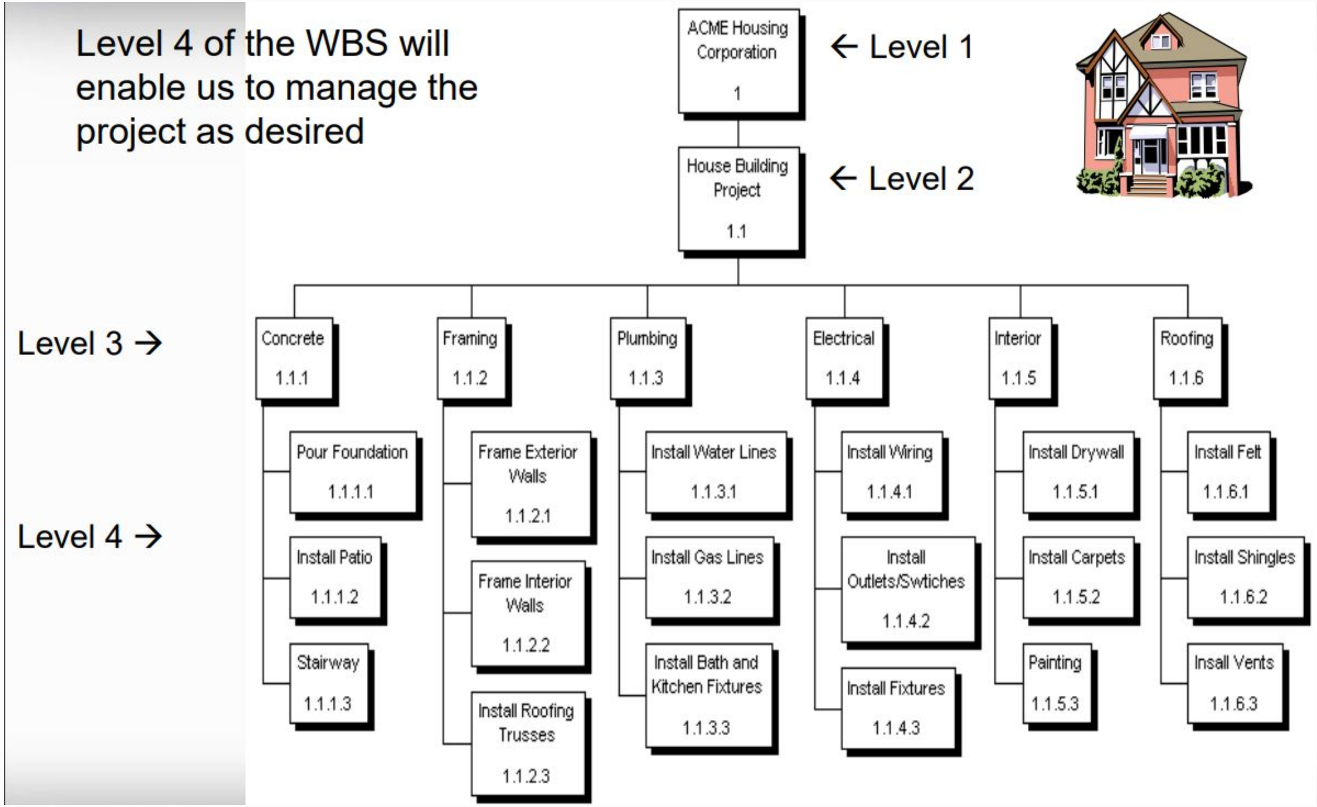


Building a WBS (Graphical Structure):

- You choose to divide **Level 3** into **six elements:** concrete, framing, plumbing, electrical, interior and roofing.
- Notice the WBS codes at **level 3**. Each is unique to the project and starts with the WBS code from the level above (11.11.1.1). Now lets look and see if another level is needed.



Building a WBS (Graphical Structure):

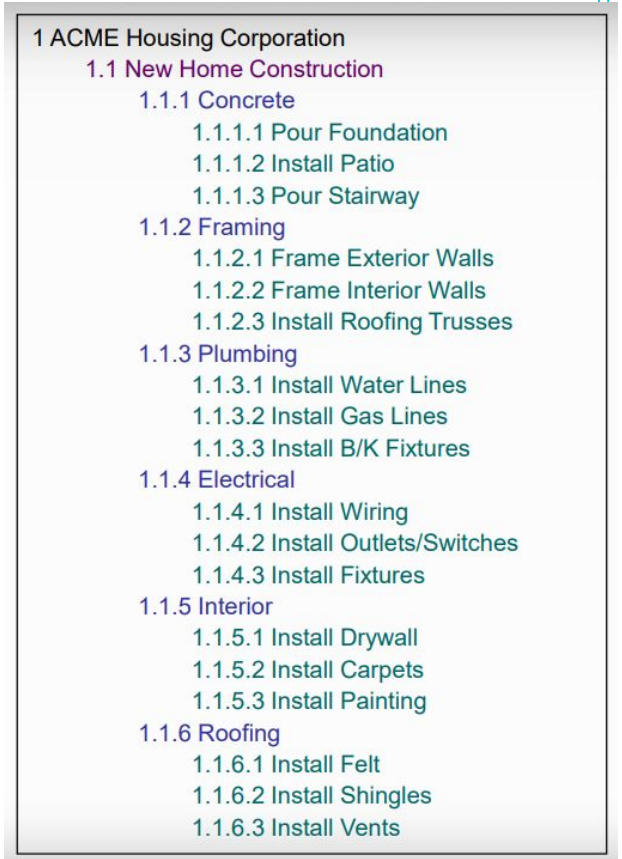


Level 4 of the WBS will enable us to manage the project as desired

- ACME Housing Corporation 1 ← Level 1
  - House Building Project 1.1 ← Level 2
    - Level 3 → Concrete 1.1.1
      - Level 4 → Pour Foundation 1.1.1.1
      - Install Patio 1.1.1.2
      - Stairway 1.1.1.3
    - Framing 1.1.2
      - Frame Exterior Walls 1.1.2.1
      - Frame Interior Walls 1.1.2.2
      - Install Roofing Trusses 1.1.2.3
    - Plumbing 1.1.3
      - Install Water Lines 1.1.3.1
      - Install Gas Lines 1.1.3.2
      - Install Bath and Kitchen Fixtures 1.1.3.3
    - Electrical 1.1.4
      - Install Wiring 1.1.4.1
      - Install Outlets/Swtiches 1.1.4.2
      - Install Fixtures 1.1.4.3
    - Interior 1.1.5
      - Install Drywall 1.1.5.1
      - Install Carpets 1.1.5.2
      - Painting 1.1.5.3
    - Roofing 1.1.6
      - Install Felt 1.1.6.1
      - Install Shingles 1.1.6.2
      - Insall Vents 1.1.6.3

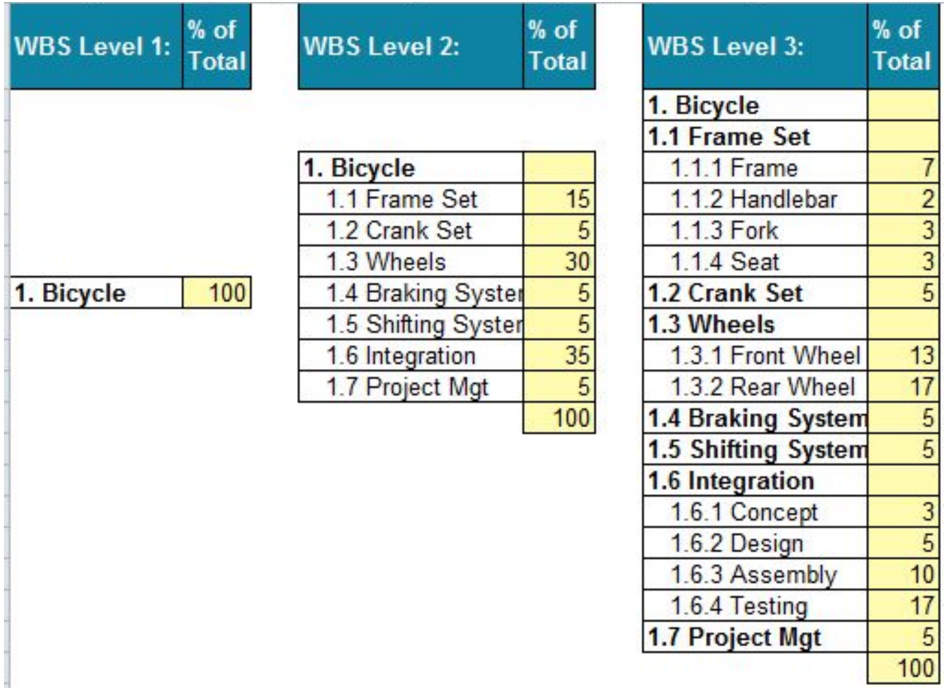
Work Breakdown Structure (WBS) Tree:

- The **graphical structure** of the WBS is an **easy** way to identify the project **components** and **relationships** of those components.
- WBS can be displayed in **another format** as well: the **Tree format**. Both formats are acceptable.
- Note the WBS codes and the structure of the Levels in the Tree format **mirror** the graphical format. The content has not changed; **only the way** the content is presented has changed.



- 1 ACME Housing Corporation
  - 1.1 New Home Construction
    - 1.1.1 Concrete
      - 1.1.1.1 Pour Foundation
      - 1.1.1.2 Install Patio
      - 1.1.1.3 Pour Stairway
    - 1.1.2 Framing
      - 1.1.2.1 Frame Exterior Walls
      - 1.1.2.2 Frame Interior Walls
      - 1.1.2.3 Install Roofing Trusses
    - 1.1.3 Plumbing
      - 1.1.3.1 Install Water Lines
      - 1.1.3.2 Install Gas Lines
      - 1.1.3.3 Install B/K Fixtures
    - 1.1.4 Electrical
      - 1.1.4.1 Install Wiring
      - 1.1.4.2 Install Outlets/Switches
      - 1.1.4.3 Install Fixtures
    - 1.1.5 Interior
      - 1.1.5.1 Install Drywall
      - 1.1.5.2 Install Carpets
      - 1.1.5.3 Install Painting
    - 1.1.6 Roofing
      - 1.1.6.1 Install Felt
      - 1.1.6.2 Install Shingles
      - 1.1.6.3 Install Vents

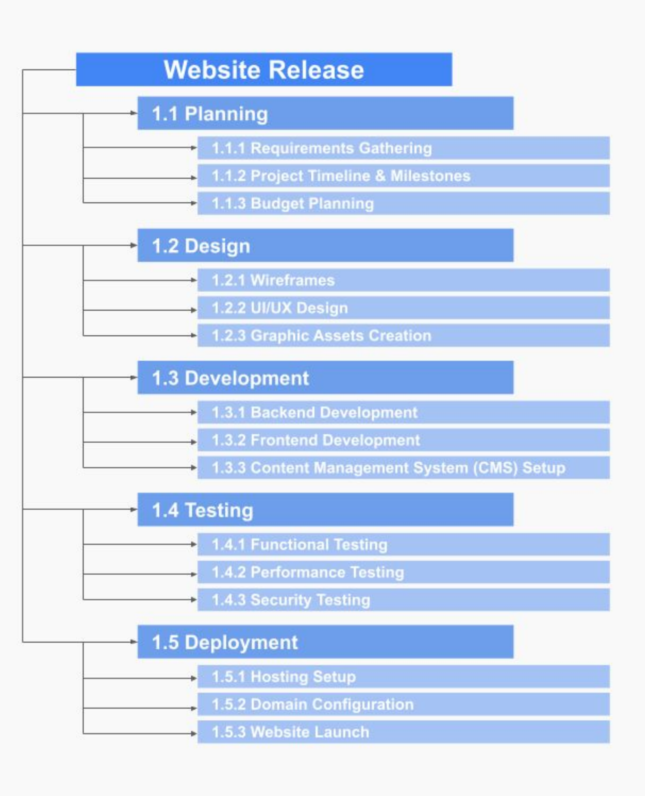
Work Breakdown Structure (WBS) Tree:



WBS Level 1:	% of Total
1. Bicycle	100

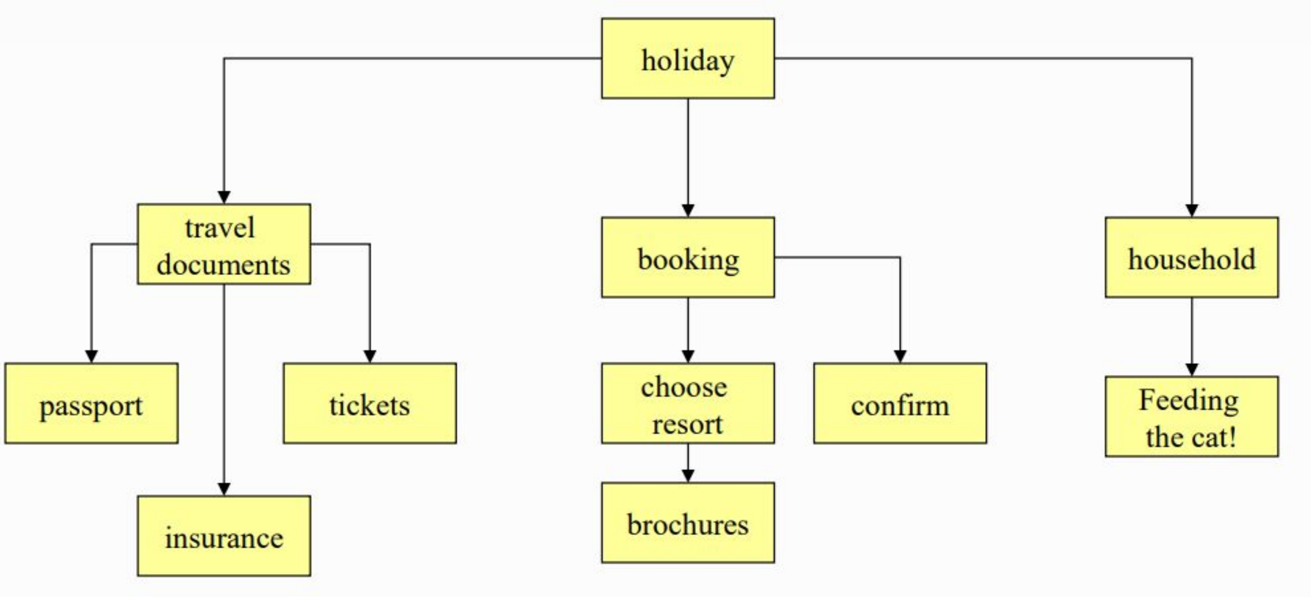
WBS Level 2:	% of Total
1. Bicycle	
1.1 Frame Set	15
1.2 Crank Set	5
1.3 Wheels	30
1.4 Braking Syster	5
1.5 Shifting Syster	5
1.6 Integration	35
1.7 Project Mgt	5
	100

WBS Level 3:	% of Total
1. Bicycle	
1.1 Frame Set	
1.1.1 Frame	7
1.1.2 Handlebar	2
1.1.3 Fork	3
1.1.4 Seat	3
1.2 Crank Set	5
1.3 Wheels	
1.3.1 Front Wheel	13
1.3.2 Rear Wheel	17
1.4 Braking System	5
1.5 Shifting System	5
1.6 Integration	
1.6.1 Concept	3
1.6.2 Design	5
1.6.3 Assembly	10
1.6.4 Testing	17
1.7 Project Mgt	5
	100



- Website Release
  - 1.1 Planning
    - 1.1.1 Requirements Gathering
    - 1.1.2 Project Timeline & Milestones
    - 1.1.3 Budget Planning
  - 1.2 Design
    - 1.2.1 Wireframes
    - 1.2.2 UI/UX Design
    - 1.2.3 Graphic Assets Creation
  - 1.3 Development
    - 1.3.1 Backend Development
    - 1.3.2 Frontend Development
    - 1.3.3 Content Management System (CMS) Setup
  - 1.4 Testing
    - 1.4.1 Functional Testing
    - 1.4.2 Performance Testing
    - 1.4.3 Security Testing
  - 1.5 Deployment
    - 1.5.1 Hosting Setup
    - 1.5.2 Domain Configuration
    - 1.5.3 Website Launch

Example of WBS : Holiday



- Booking:
  - get brochures
  - choose resort
  - make booking
  - confirm booking
- Travel documents:
  - check passport
  - book tickets
  - get insurance
- Household:
  - feeding the cat!

Need / Benefits of a Work Breakdown Structure (WBS):

- ❖ **Improved Organization:** Provides a **structured framework** for managing tasks, enhancing resource allocation, budgeting, and stakeholder management.
- ❖ **Enhanced Clarity:** Clearly defined tasks **improve communication**, minimize conflicts, and ensure team alignment.
- ❖ **Better Time Management:** Helps create **accurate estimates**, track progress, and maximize efficiency while controlling costs.
- ❖ **Early Risk Identification:** Identifies potential risks like **bottlenecks early**, allowing for timely mitigation strategies to meet project milestones effectively.
- ❖ **Enhances Cost Control:** Allows more **accurate cost estimates** and better cost tracking.

▶ Network Planning Models 2.3

Network Planning Model:

Network Planning Model are the techniques used to represent the **sequence** and **interdependence** of project activities. These are the techniques for visualizing and analyzing the task involved in a project.

- ➔ It represents **project as a network of activities** connected by dependencies, showing the order in which task must be completed and how delays in one task can impact others.
- ➔ Scheduling activities model the projects activities and their relationship as a “**Network**”.
- ➔ In Network **time flows from left to right**.
- ➔ These techniques were developed in **1950’s**.
- ➔ There are **2 best techniques** of Network Planning :
  - ◆ CPM (Critical Path Method)
  - ◆ PERT (Program Evaluation and Review Technique)

Network Planning Model:

The most commonly used models are:

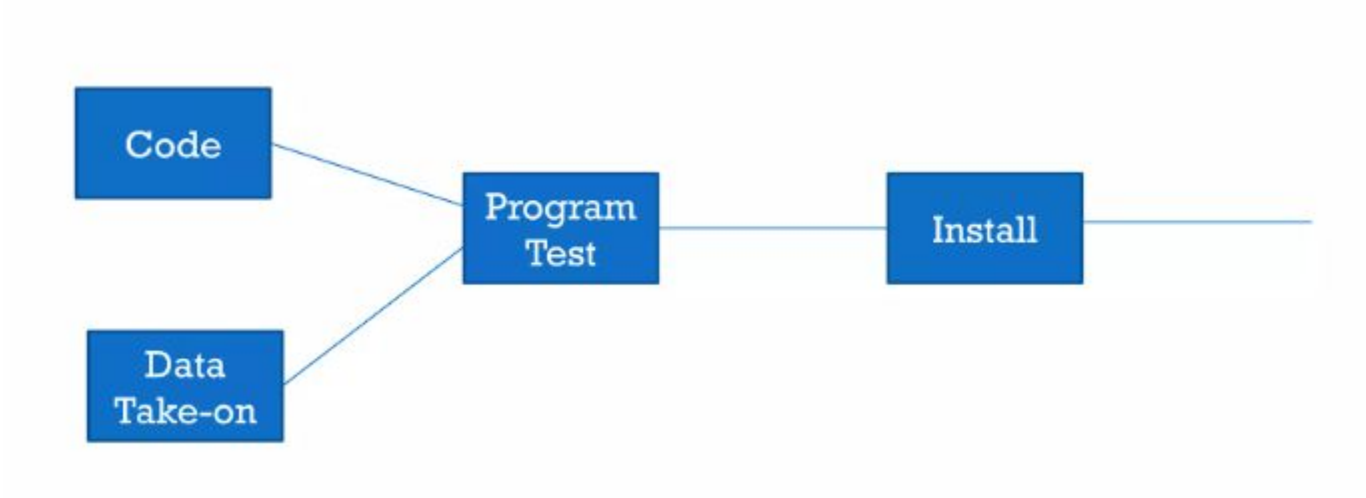
- **Program Evaluation and Review Technique (PERT):** A project management tool used to analyze the time required to complete tasks and identify the probability of meeting deadlines.
  - To manage projects with uncertain activity durations.
- **Critical Path Method (CPM):** A project management technique that identifies the longest sequence of dependent activities (critical path) to determine the minimum project duration.
  - To identify time-critical tasks that directly impact the project schedule.
- Both method uses "**activity-on-arrow**" approach.
- Activities are drawn as **arrows joining circles**, or nodes, which shows the **start and completion time** of the activities.

Precedence Network:

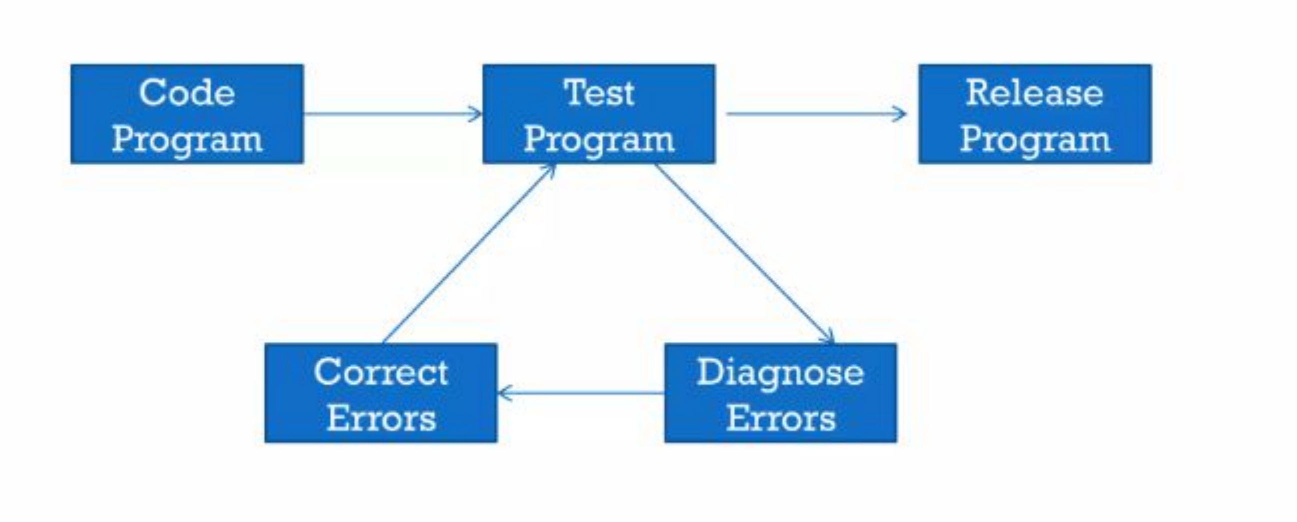
- Recently used, Popular one currently.
- It uses “**activity on node**” networks.
- Activities are represented as **Nodes**.
- Link between the nodes are represented as **sequencing requirements.**
-

- **Rules: Constructing precedence networks**
  - A project network should have only one start node
  - A project network should have only one end node
  - A node has duration
  - A node represents an activity and, in general, activities take time to execute.
  - Links normally have no duration
  - Precedents are the immediate preceding activities
  - Time moves from left to right
  - A network may not contain Loops
  - A network should not contain dangles.

Fragment of Precedence Network:

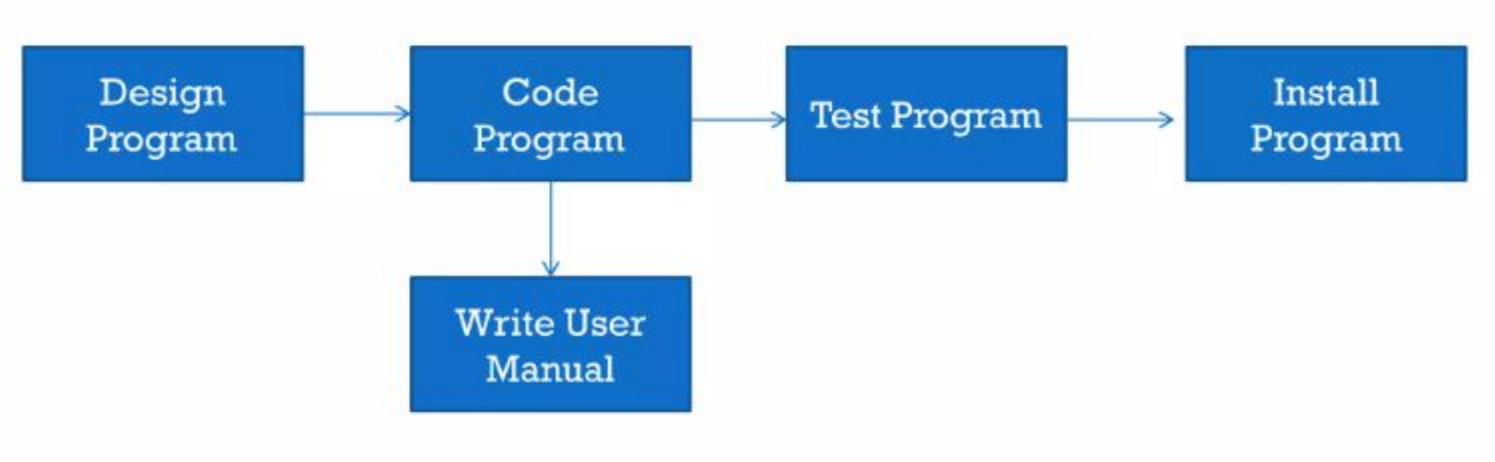


A Loop represents an impossible sequence:



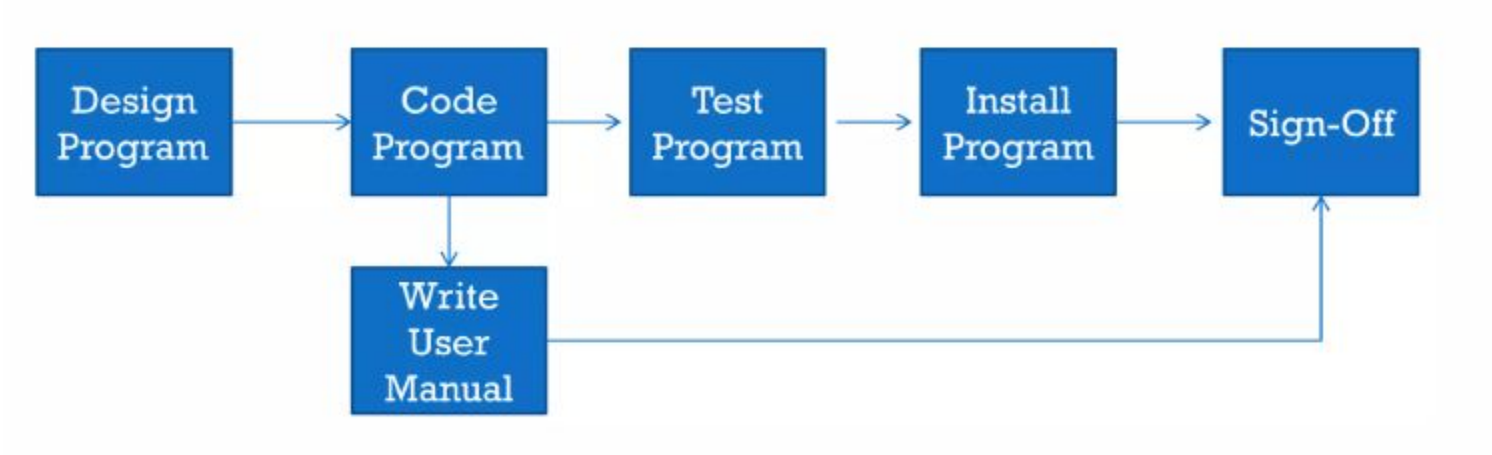
A loop is an error in that it represents a situation that cannot occur in practice. While loops, in the sense of iteration, may occur in practice, they cannot be directly represented in a project network.

A Dangle:



A network should not contain dangles. A dangling activity such as 'Write user manual' in Figure should not exist as it is likely to lead to errors in sub-sequent analysis

Resolving the Dangle:



Redraw the network with a final completion activity — which, at least in this case, is probably a more accurate.

- **Representing Lagged Activities:**
  - We might come across situations where we wish to undertake two activities in parallel so long as there is a lag between the two.
  - Eg. We would designate an activity "test and document amendments which would make it impossible to show that amendment recording could start say 1 day after testing has begun and finishing a little after the completion of testing.
  - The figure indicates that the document amendments can start one day after the start of prototype testing and will be completed two days after prototype testing us completed.

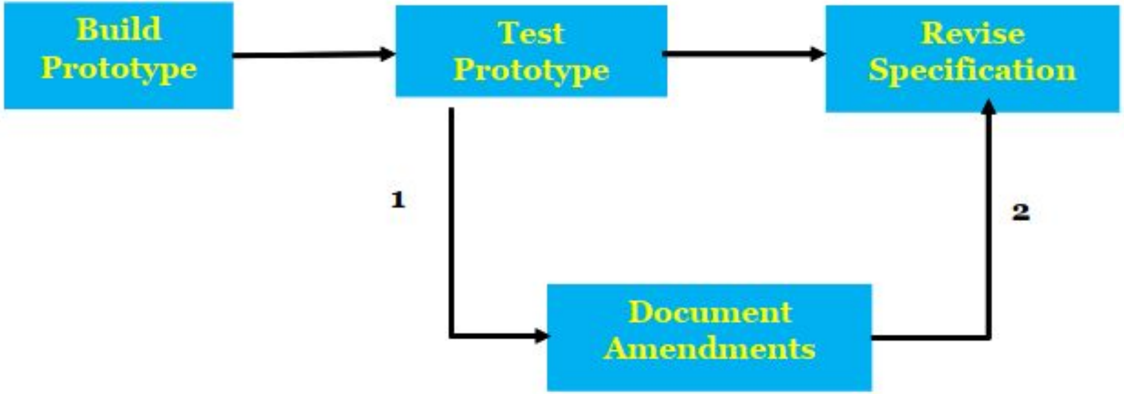
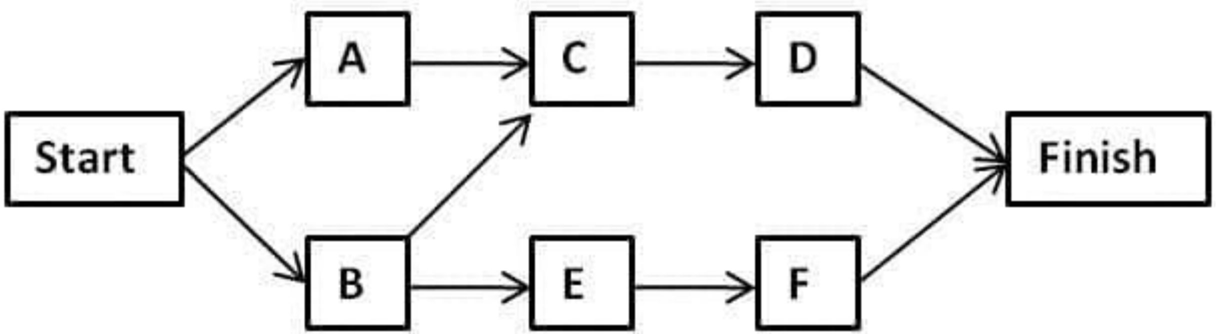


Fig. Indicating Lags

PDM:

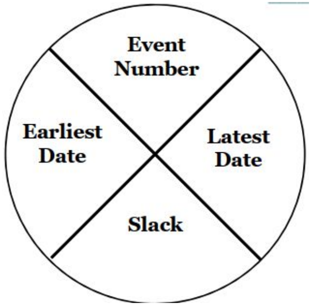


Precedence Diagram Method (PDM)

Labelling Conventions:

- There are a number of differing conventions that have been adopted for entering information on an activity-on-node network.
- One of the more common conventions for labelling nodes, and the one adopted here, is shown on the left.
- The activity label is usually a code developed to uniquely identify the activity and may incorporate a project code. The activity description will normally be a brief activity name such as 'Test take-on module'.

<table>
<tr><td>Earliest start</td><td>Duration</td><td>Earliest finish</td></tr>
<tr><td colspan="3">Activity label, activity description</td></tr>
<tr><td>Latest start</td><td>Float</td><td>Latest finish</td></tr>
</table>

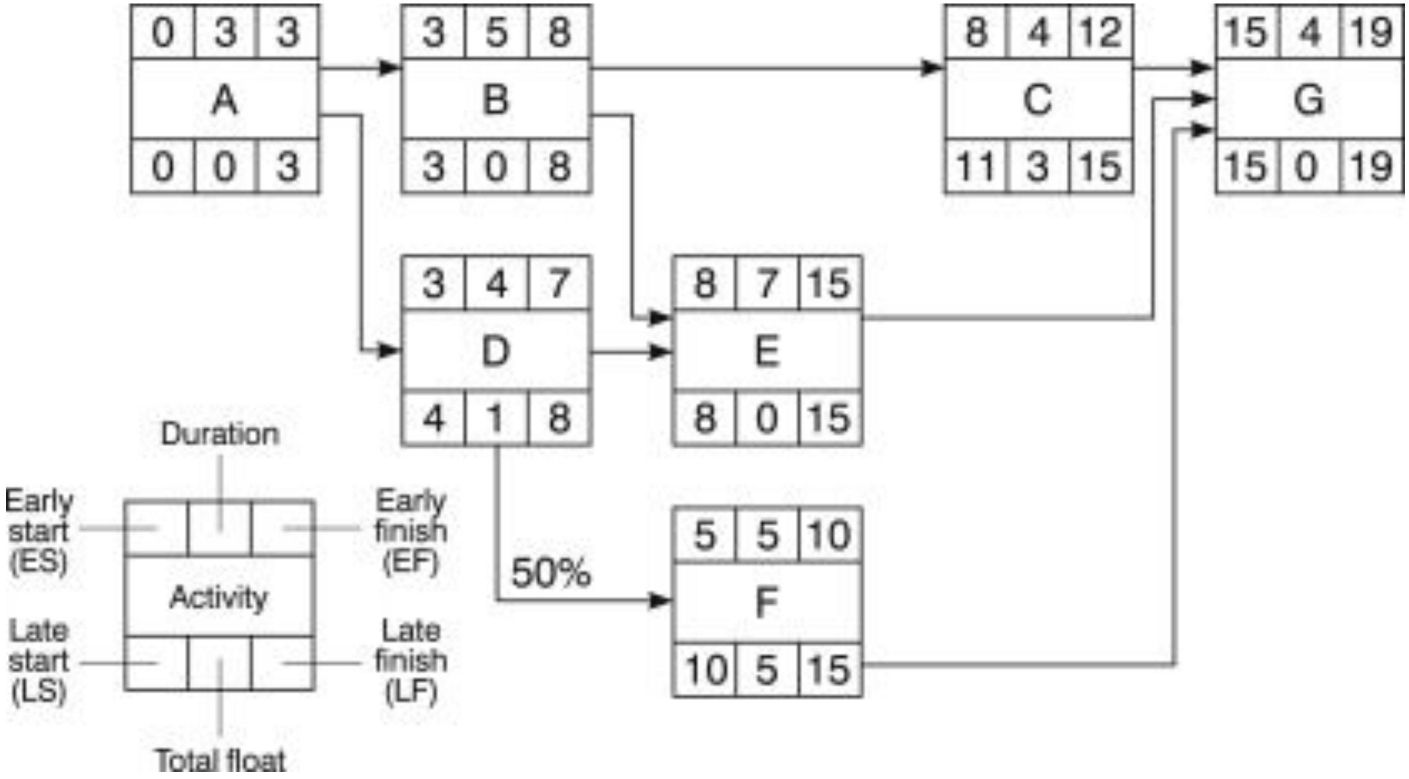


<table>
<tr><td colspan="2">Activity Label</td><td colspan="2">Duration</td></tr>
<tr><td>Earliest Start</td><td colspan="2" rowspan="2">Activity Description</td><td>Earliest Finish</td></tr>
<tr><td>Latest Start</td><td>Latest Finish</td></tr>
<tr><td colspan="2">Activity Span</td><td colspan="2">Float</td></tr>
</table>

Adding Time Dimension:

- Each activity has an estimate of its duration.
- Forward Pass:
  - Earliest date at which activities may commence and project be completed.
- Backward Pass:
  - Latest start dates for activities and critical path.

PDM with Forward Pass & Backward Pass:



AON & AOA

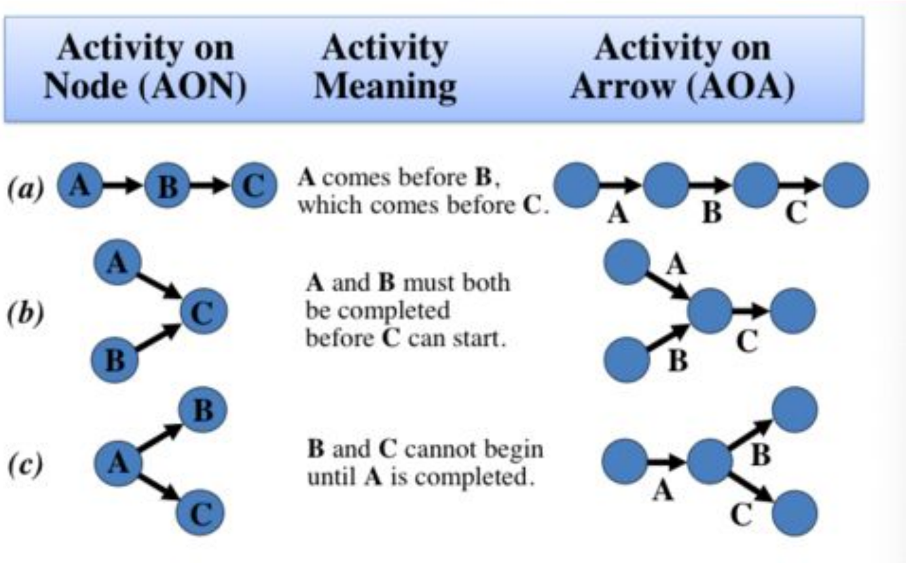
Activity on Node (Box):

- Nodes **represent activities**, and arrows show **precedence relationships**
- Each **activity is represented by a box.**
- The **activity description** is written in the box.
- Activity consume time, and their description usually starts with a verb.
- Each activity is represented by **one and only one box.**
- Each box is assigned a **unique activity number.**
- Activities have a **precedential relationship.**
- Some activities may be done concurrently.

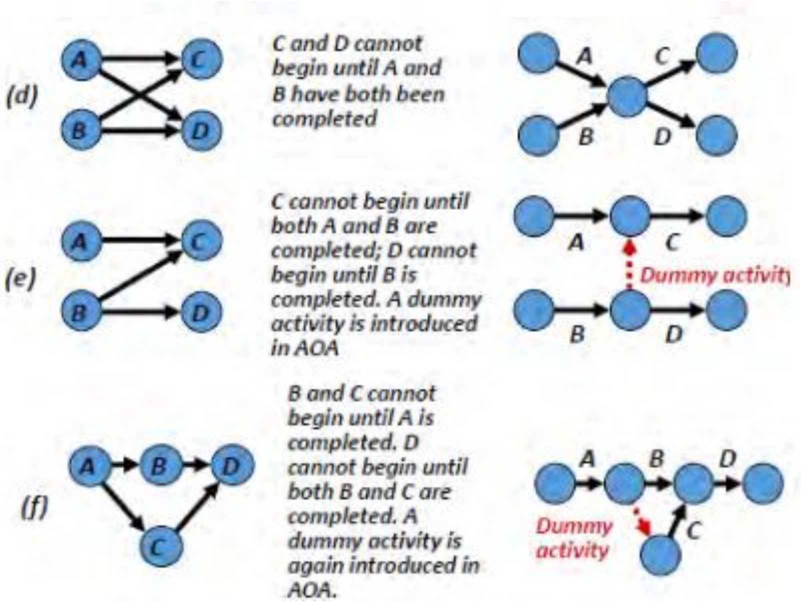
Activity on Arrow:

- Arrows represent activities and nodes are events for points in time
- Each activity is represented by an arrow.
- The activity description is written above the arrow.
- Each activity is represented by one and only one arrow
- The tail of the arrow designates the start of the activity.
- The head of the arrow designates the completion of the activity.
- The length and slope of the arrow are in no way indicative of the activity’s duration or importance

Comparison of AON and AOA Networks:

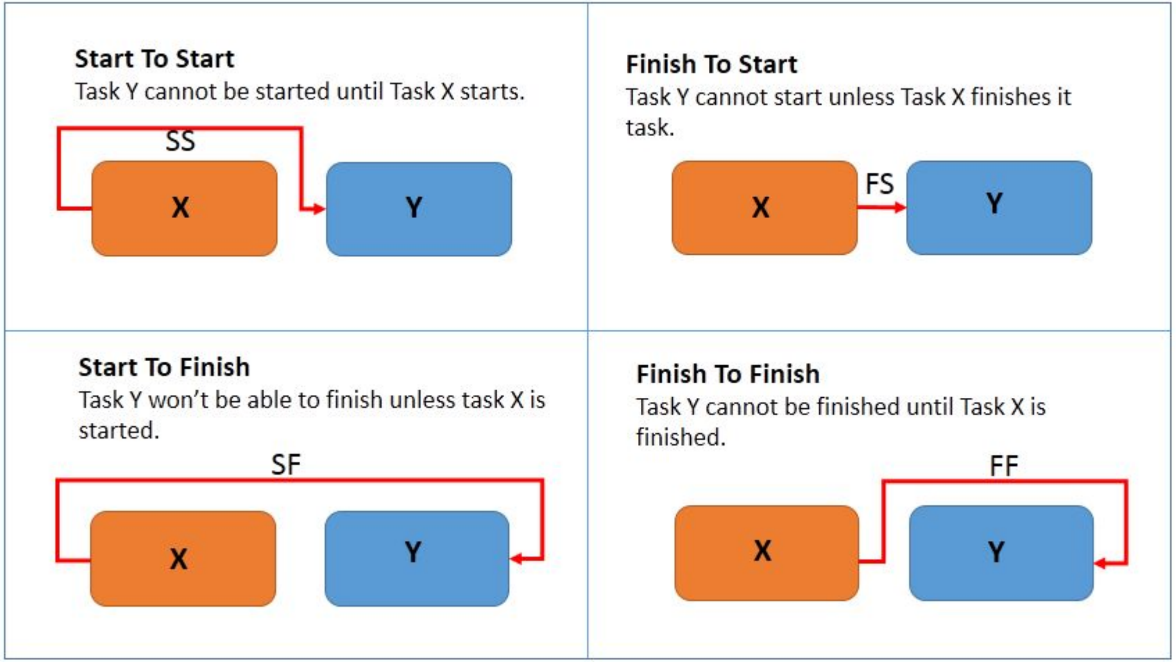


	Activity on Node (AON)	Activity Meaning	Activity on Arrow (AOA)
(a)		A comes before B, which comes before C.	
(b)		A and B must both be completed before C can start.	
(c)		B and C cannot begin until A is completed.	



(d)		C and D cannot begin until A and B have both been completed	
(e)		C cannot begin until both A and B are completed; D cannot begin until B is completed. A dummy activity is introduced in AOA	
(f)		B and C cannot begin until A is completed. D cannot begin until both B and C are completed. A dummy activity is again introduced in AOA.	

Task Dependency Relationship:



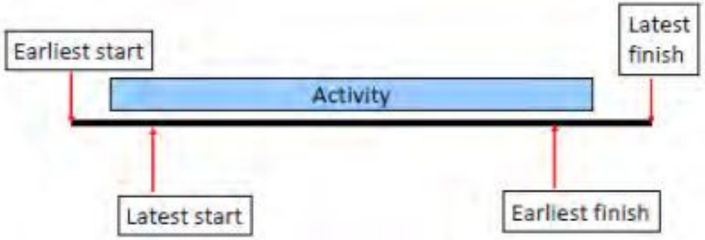
Start To Start
Task Y cannot be started until Task X starts.

Finish To Start
Task Y cannot start unless Task X finishes it task.

Start To Finish
Task Y won’t be able to finish unless task X is started.

Finish To Finish
Task Y cannot be finished until Task X is finished.

Start & Finish Times:



- **Earliest start time (ES)** is the earliest time at which a particular activity can begin.
- **Earliest finish time (EF)** is the earliest time by which a particular activity can be completed.
  - **EF = ES + Duration Estimate**
- **Latest finish time (LF)** is the latest time by which a particular activity must be finished in order for the entire project to be completed by its required completion time, calculated on the basis of the project’s required completion time and the duration estimates for succeeding activities.
- **Latest start time (LS)** is the latest time by which a particular activity must be started in order for the entire project to be completed by its required completion time, calculated by subtracting the activity’s duration estimate from the activity’s latest finish time:
  - **LS=LF– Duration Estimate**

Critical Path Method

2.3.1

Critical Path Method:

- A project management technique that identifies the **longest sequence of dependent activities** (critical path) to determine the minimum project duration.
- **History:**
  - Developed **by Morgan R. Walker** and **James E. Kelley** in the late 1950s while working for DuPont.
  - First used in chemical plant construction projects.
- **Why It's Used?**
  - To **identify time-critical tasks** that directly **impact** the project schedule.
  - Ensures efficient resource utilization and helps in **delay management.**

Critical Path Method:

- **Key Features:**
  - **Focus on Critical Path:** Tasks on the critical path must be **completed on time** to avoid delays.
  - **Time Optimization:** Helps calculate earliest and latest start/finish times for activities.
  - **Dependency Analysis:** Shows task relationships (e.g., finish-to-start).
- **Critical Path:**
  - The **path with the largest time** is called the critical path and the activities along this path are called the critical activities or bottleneck activities.
  - Any delay in an activity on the critical path directly **impacts the project completion date**.
- **Critical Event:**
  - An event that has **zero slack** time.

▶ Critical Path Method:

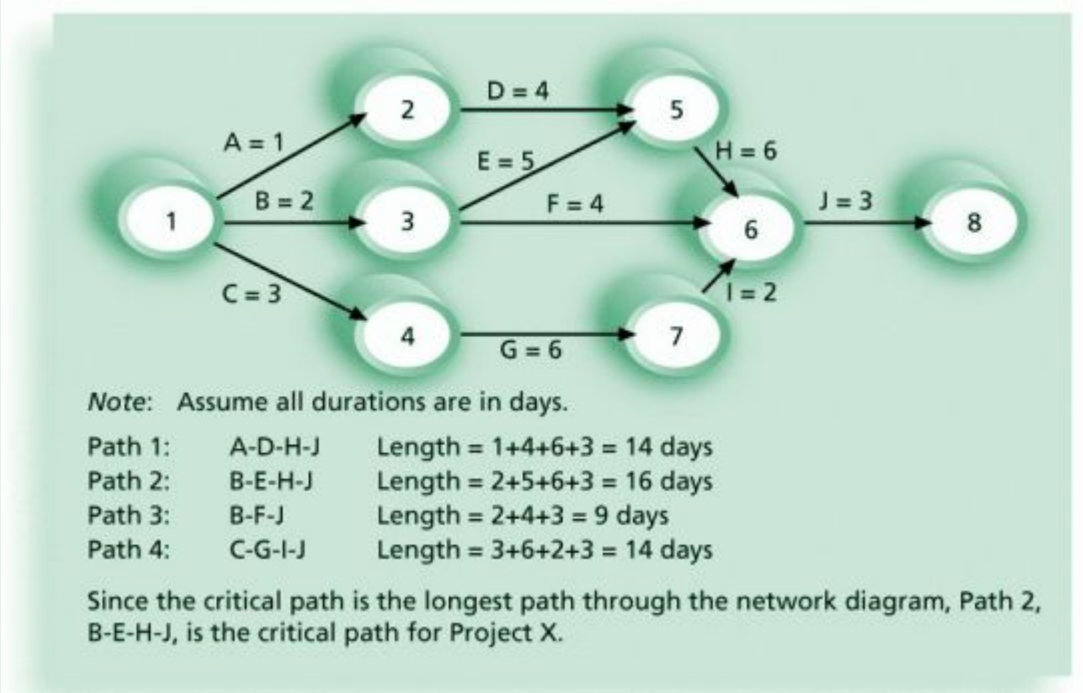
Key Elements of Critical Path:

- ❖ **Critical Activities:**
  - ➢ Tasks that must be completed on time to avoid delays in the project.
- ❖ **Dependencies:**
  - ➢ Relationships between activities (e.g., finish-to-start).
- ❖ **Duration:**
  - ➢ Total time required for the critical path activities to complete.

How to Determine the Critical Path?

- ❖ **List All Activities:** Include task durations and dependencies.
- ❖ **Create a Network Diagram:** Visualize tasks and their relationships.
- ❖ **Identify Paths:** Calculate the total duration for each sequence of activities.
- ❖ **Select the Longest Path:** The longest path is the critical path.

Critical Path Method- Example:

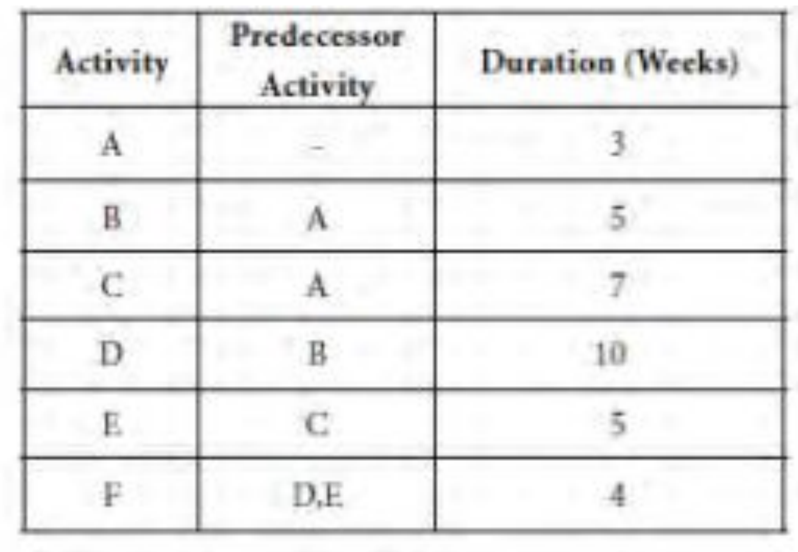


Note: Assume all durations are in days.

Path 1:	A-D-H-J	Length = 1+4+6+3 = 14 days
Path 2:	B-E-H-J	Length = 2+5+6+3 = 16 days
Path 3:	B-F-J	Length = 2+4+3 = 9 days
Path 4:	C-G-I-J	Length = 3+6+2+3 = 14 days

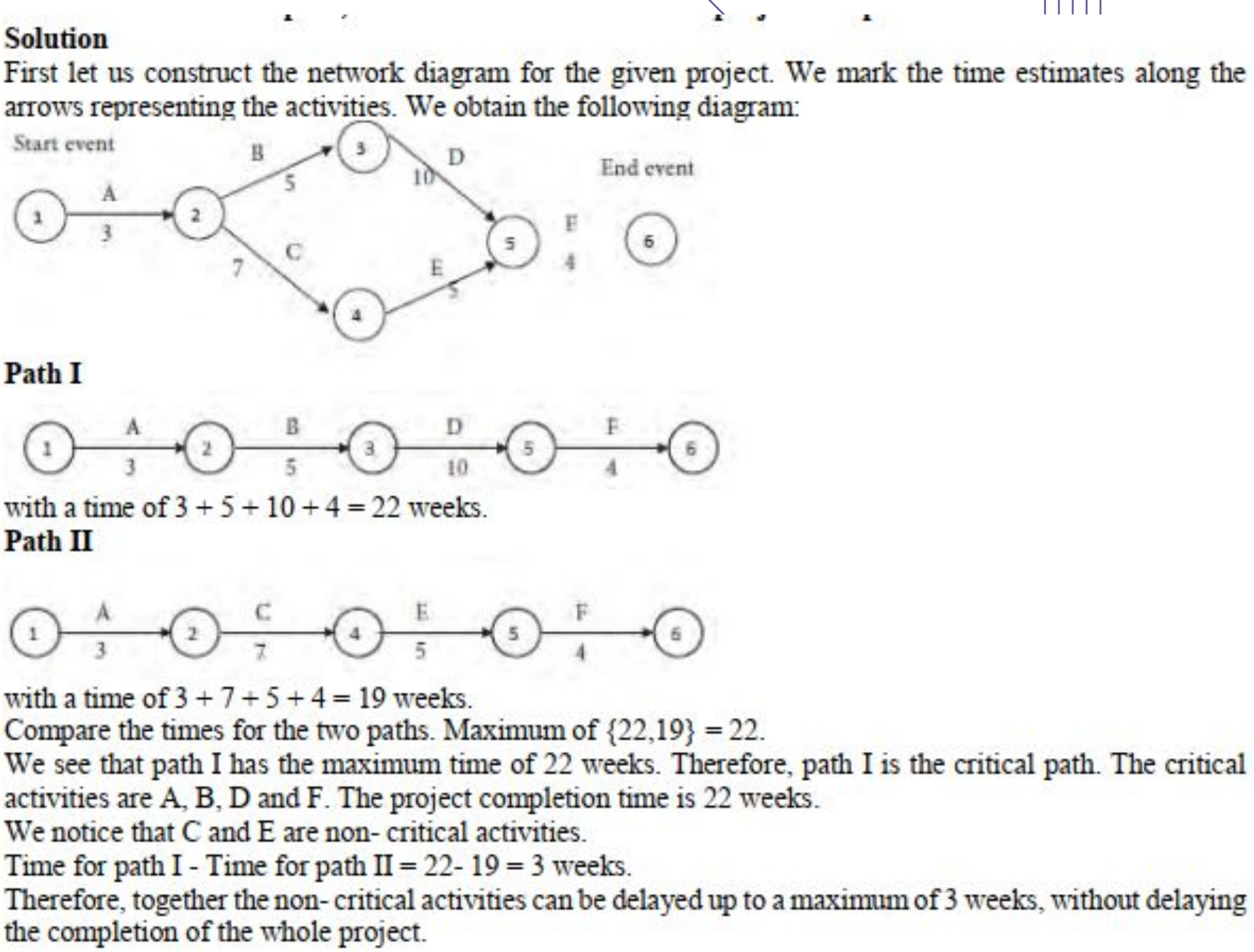
Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.

Q-1: Determine the critical path, the critical activities and the project completion time



Activity	Predecessor Activity	Duration (Weeks)
A	–	3
B	A	5
C	A	7
D	B	10
E	C	5
F	D,E	4

Solution:



Solution

First let us construct the network diagram for the given project. We mark the time estimates along the arrows representing the activities. We obtain the following diagram:

Start event
End event
1
A
3
2
B
5
3
D
10
C
7
4
E
5
5
F
4
6

Path I

1
A
3
2
B
5
3
D
10
5
F
4
6

with a time of 3 + 5 + 10 + 4 = 22 weeks.

Path II

1
A
3
2
C
7
4
E
5
5
F
4
6

with a time of 3 + 7 + 5 + 4 = 19 weeks.

Compare the times for the two paths. Maximum of {22,19} = 22.

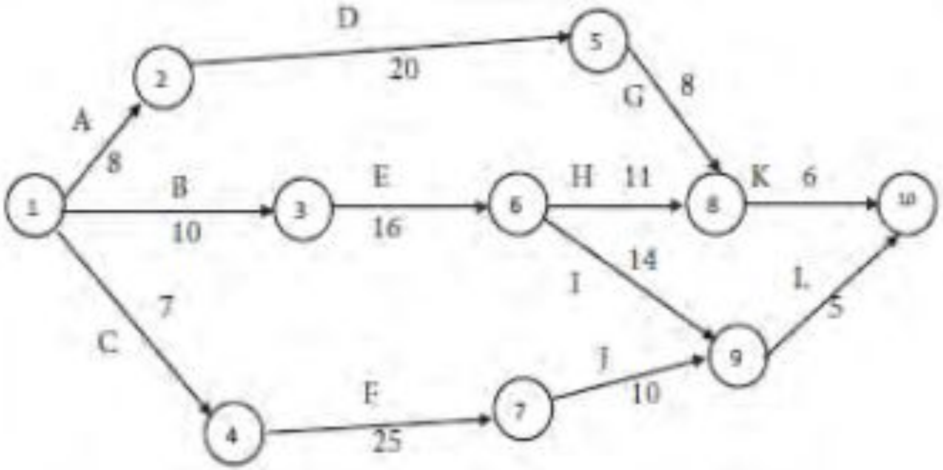
We see that path I has the maximum time of 22 weeks. Therefore, path I is the critical path. The critical activities are A, B, D and F. The project completion time is 22 weeks.

We notice that C and E are non- critical activities.

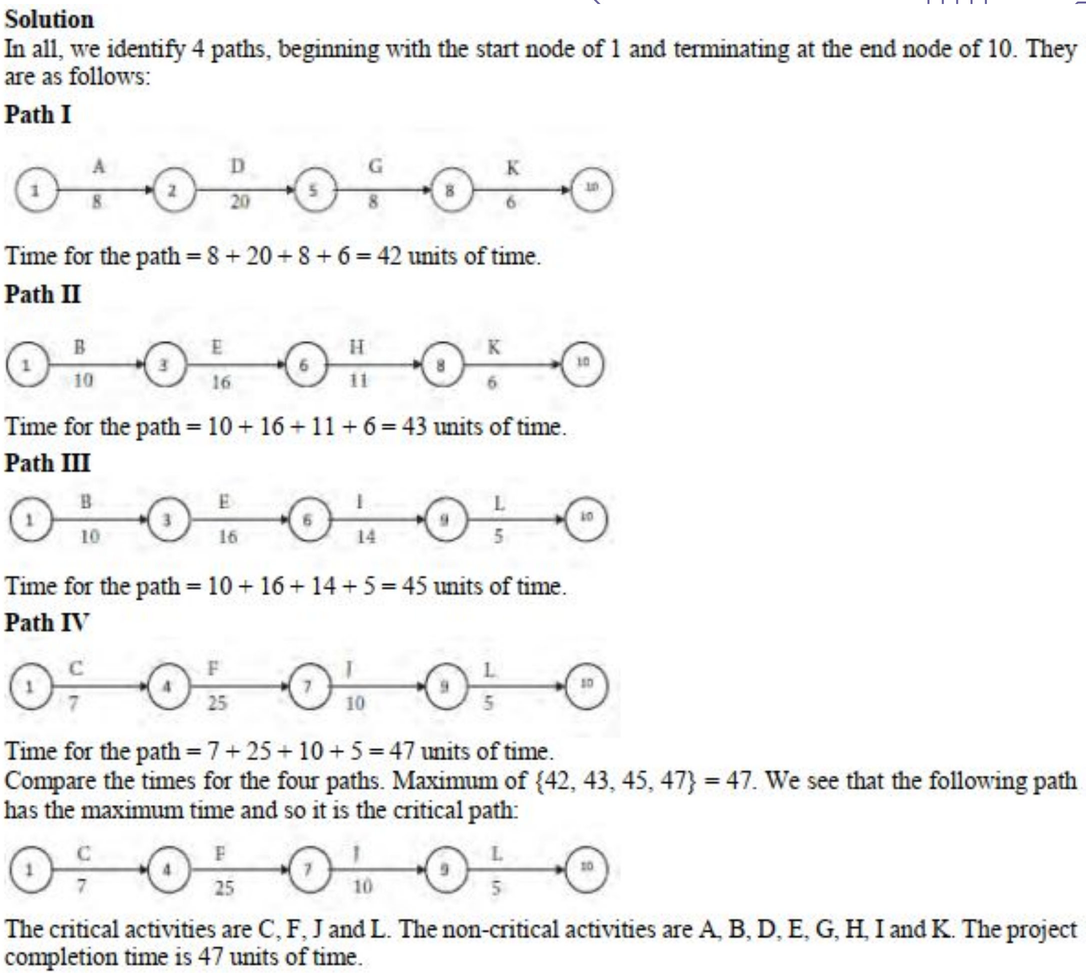
Time for path I - Time for path II = 22- 19 = 3 weeks.

Therefore, together the non- critical activities can be delayed up to a maximum of 3 weeks, without delaying the completion of the whole project.

- **Q-2: Find out the completion time and the critical activities for the following project:**



Solution:



Solution

In all, we identify 4 paths, beginning with the start node of 1 and terminating at the end node of 10. They are as follows:

Path I

1
A
8
2
D
20
5
G
8
8
K
6
10

Time for the path = 8 + 20 + 8 + 6 = 42 units of time.

Path II

1
B
10
3
E
16
6
H
11
8
K
6
10

Time for the path = 10 + 16 + 11 + 6 = 43 units of time.

Path III

1
B
10
3
E
16
6
I
14
9
L
5
10

Time for the path = 10 + 16 + 14 + 5 = 45 units of time.

Path IV

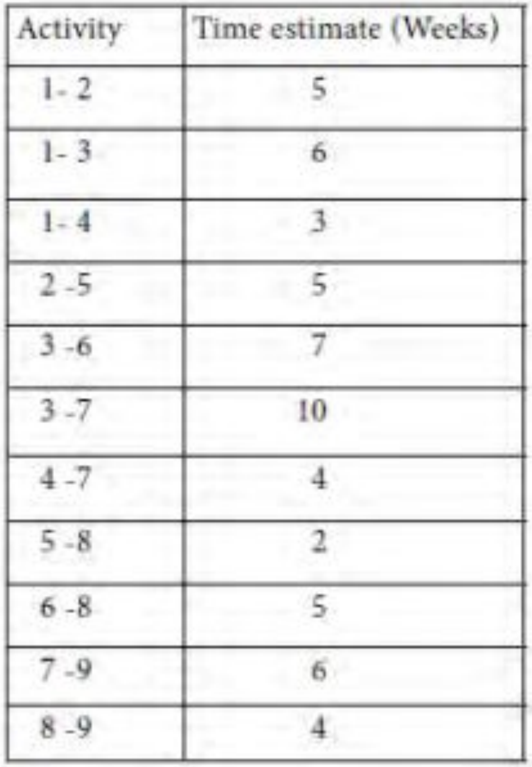
1
C
7
4
F
25
7
J
10
9
L
5
10

Time for the path = 7 + 25 + 10 + 5 = 47 units of time.

Compare the times for the four paths. Maximum of {42, 43, 45, 47} = 47. We see that the following path has the maximum time and so it is the critical path:

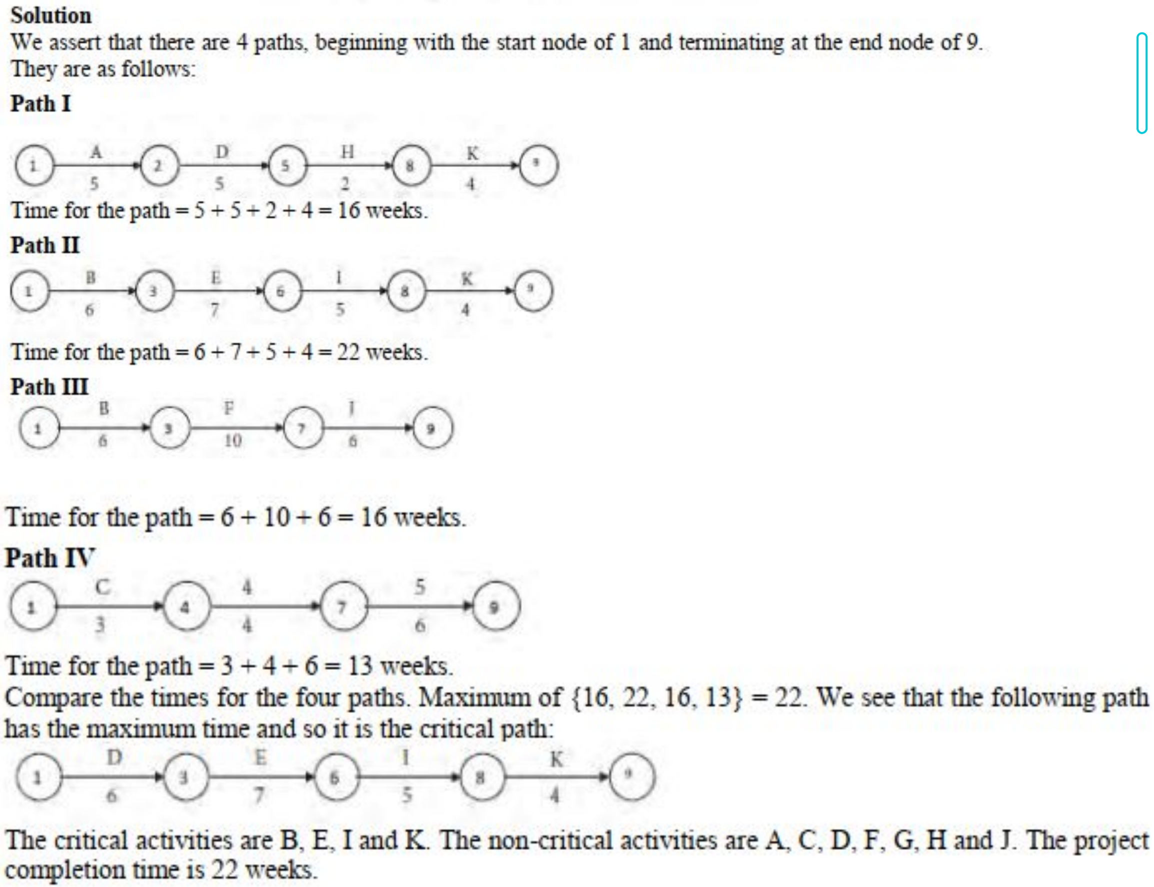
The critical activities are C, F, J and L. The non-critical activities are A, B, D, E, G, H, I and K. The project completion time is 47 units of time.

Q-3: Draw the network diagram and determine the critical path for the following project:



Activity	Time estimate (Weeks)
1- 2	5
1- 3	6
1- 4	3
2 -5	5
3 -6	7
3 -7	10
4 -7	4
5 -8	2
6 -8	5
7 -9	6
8 -9	4

Solution:



Solution

We assert that there are 4 paths, beginning with the start node of 1 and terminating at the end node of 9. They are as follows:

Path I

1
A
5
2
D
5
5
H
2
8
K
4
9

Time for the path = 5 + 5 + 2 + 4 = 16 weeks.

Path II

1
B
6
3
E
7
6
I
5
8
K
4
9

Time for the path = 6 + 7 + 5 + 4 = 22 weeks.

Path III

1
B
6
3
F
10
7
J
6
9

Time for the path = 6 + 10 + 6 = 16 weeks.

Path IV

1
C
3
4
4
4
7
5
6
9

Time for the path = 3 + 4 + 6 = 13 weeks.

Compare the times for the four paths. Maximum of {16, 22, 16, 13} = 22. We see that the following path has the maximum time and so it is the critical path:

The critical activities are B, E, I and K. The non-critical activities are A, C, D, F, G, H and J. The project completion time is 22 weeks.

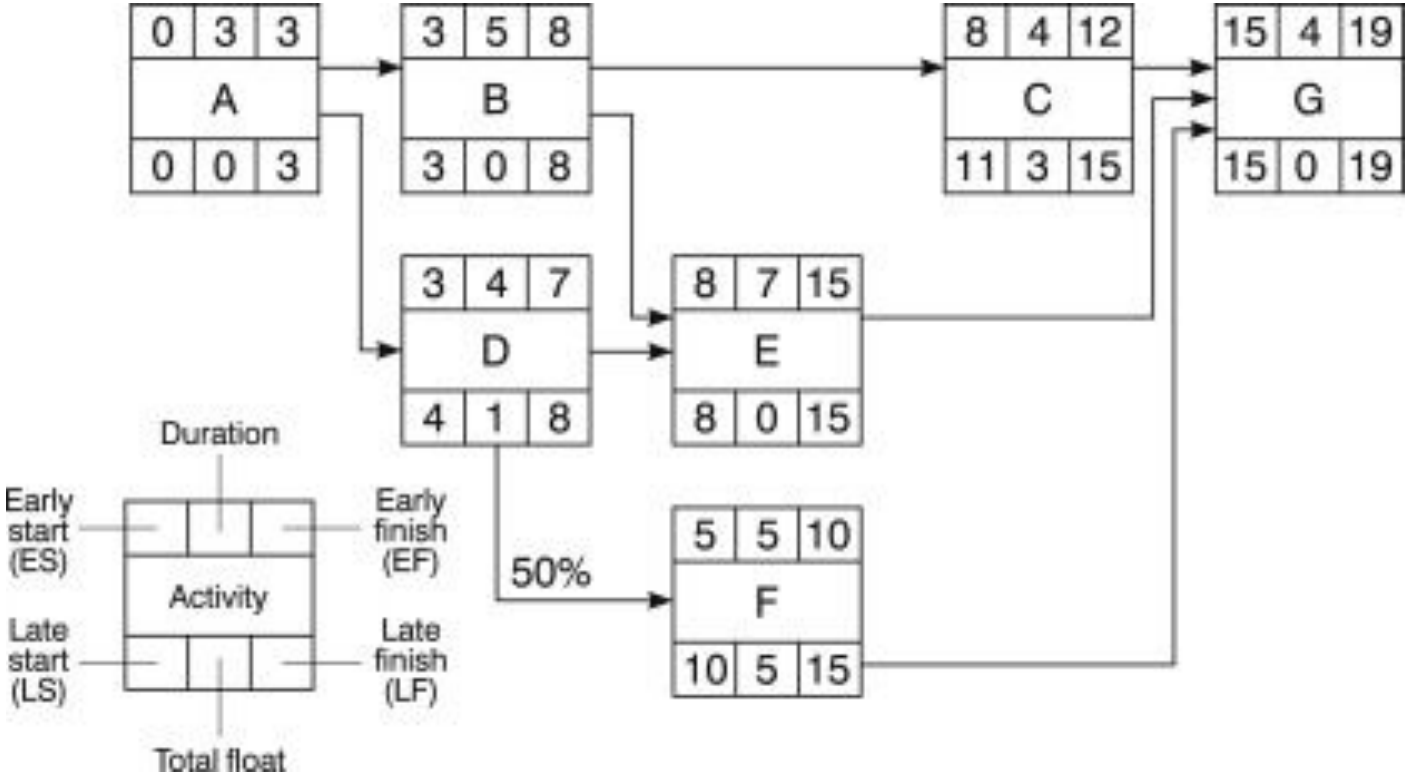
Forward & Backward Pass

2.6

Adding Time Dimension:

- Each activity has an estimate of its duration.
- **Forward Pass:**
  - Calculate the earliest start date of the activities.
  - To calculate the project completion date.
- **Backward Pass:**
  - Calculate the latest start date for activities
  - Identify the critical path from the graph.
- **Float** is equal to the difference between the **earliest start and latest start** or **earliest finish and latest finish.** **Float/Slack = LS-ES or LF-EF**
- Activities with **zero slack** are on the critical path, meaning delays in these tasks will directly impact the project timeline.

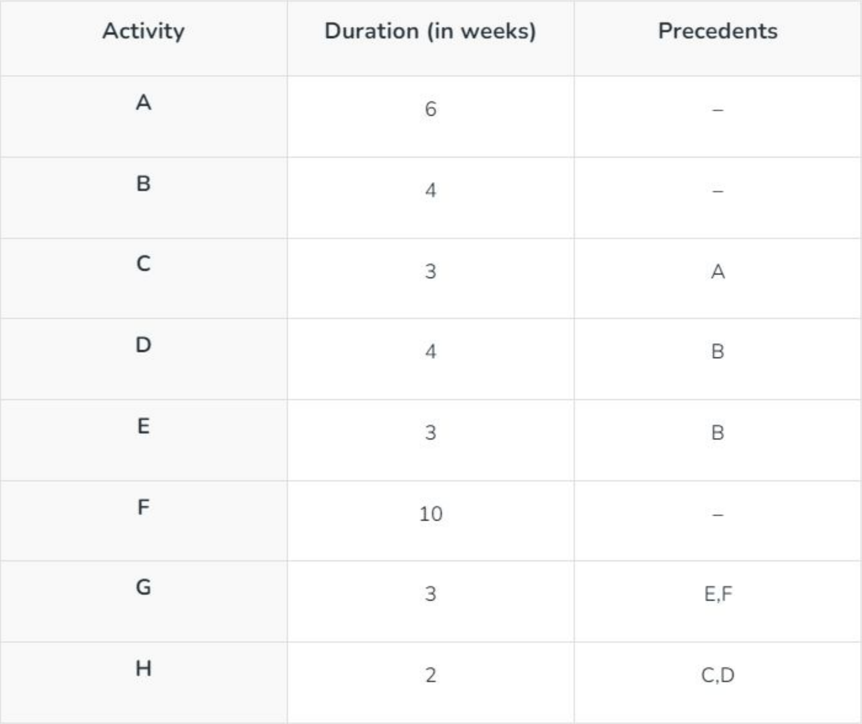
PDM with Forward Pass & Backward Pass:



Summary: How to find the critical path in a project?

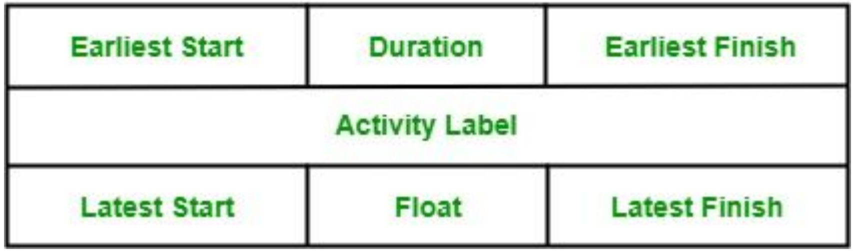
- Step 1: Identify all tasks required to complete the project
- Step 2: Determine the sequence of tasks
- Step 3: Estimate the duration of each task
- Step 4: Draw a network diagram
- Step 5: Identify the critical path
- Step 6: Calculate the float
- Step 7: Monitor the critical path

- **Q. The table given below contains the activity label, its respective duration (in weeks), and its precedents. We will use the critical path method to find the critical path and activities of this project.**



Activity	Duration (in weeks)	Precedents
A	6	–
B	4	–
C	3	A
D	4	B
E	3	B
F	10	–
G	3	E,F
H	2	C,D

Node Representation:

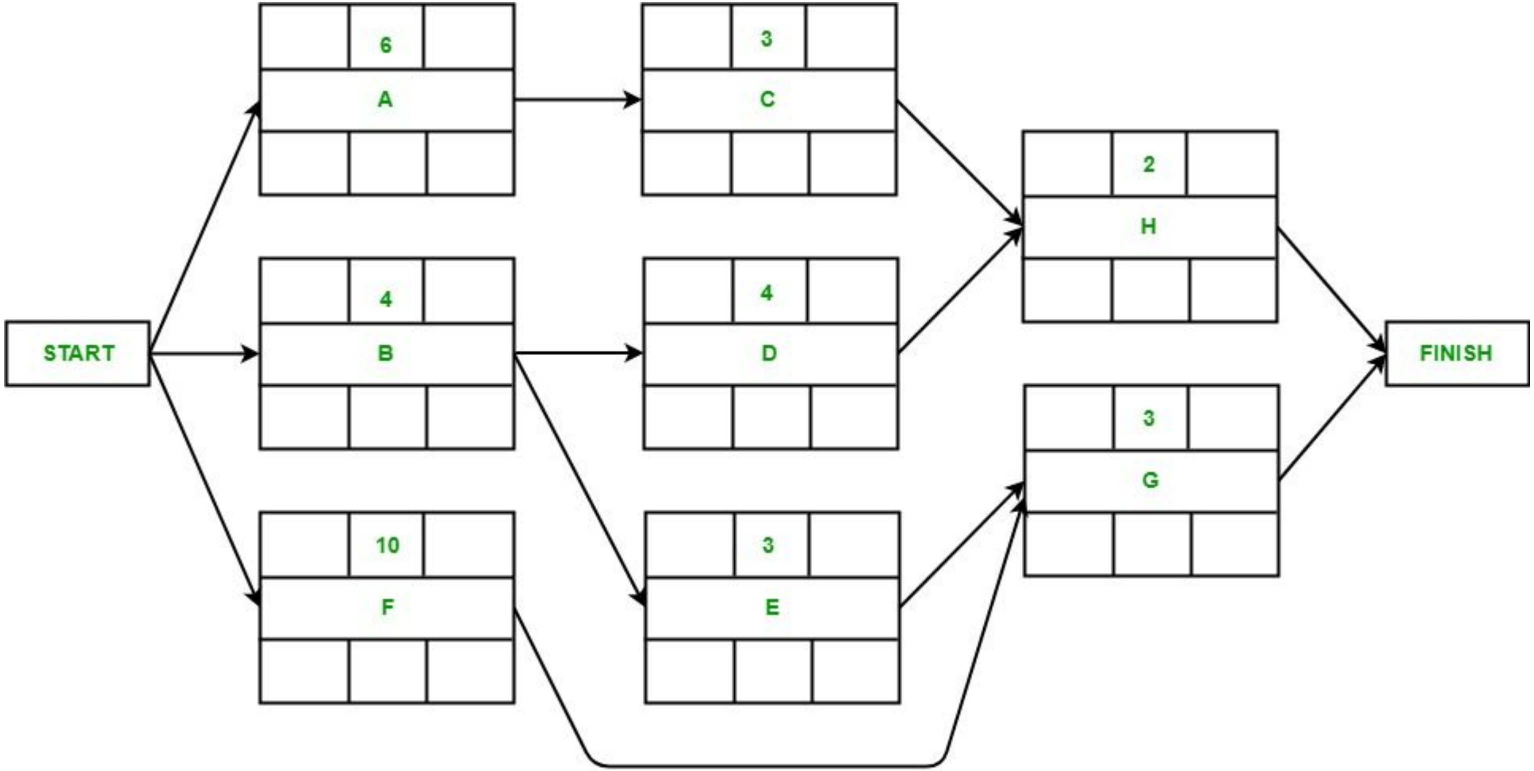


Earliest Start	Duration	Earliest Finish
Activity Label		
Latest Start	Float	Latest Finish

Here,

- ❖ **Activity label** is the name of the activity represented by that node.
- ❖ **Earliest Start** is the date or time at which the activity can be started at the earliest.
- ❖ **Earliest Finish** is the date or time at which the activity can be completed at the earliest.
- ❖ **Latest Start** is the date or time at which the activity can be started at the latest.
- ❖ The **Latest Finish** is the date or time at which the activity can be finished at the latest.
- ❖ **Float** is equal to the difference between the earliest start and latest start or earliest finish and latest finish.

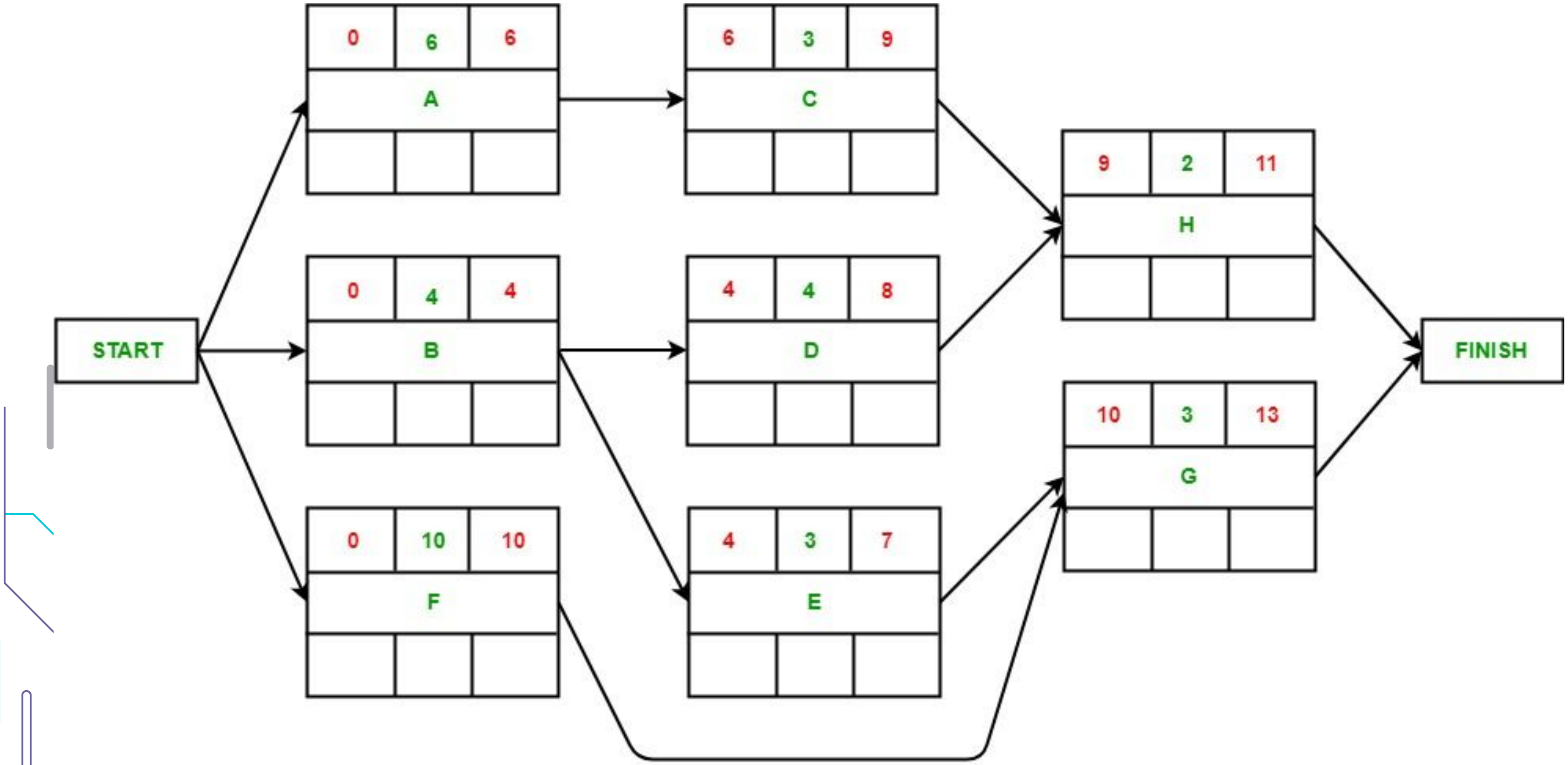
- **The precedence network would look like:**



CPM-Forward Pass

- During the forward pass, **earliest dates** are recorded as they are calculated.
- **The forward pass rule:**
  - The **earliest start date** for an activity is the **earliest finish date** for the preceding activity.
  - Where there are **more than one** immediately preceding activity we take the ***latest*** of the ***earliest finish dates*** for those activities.

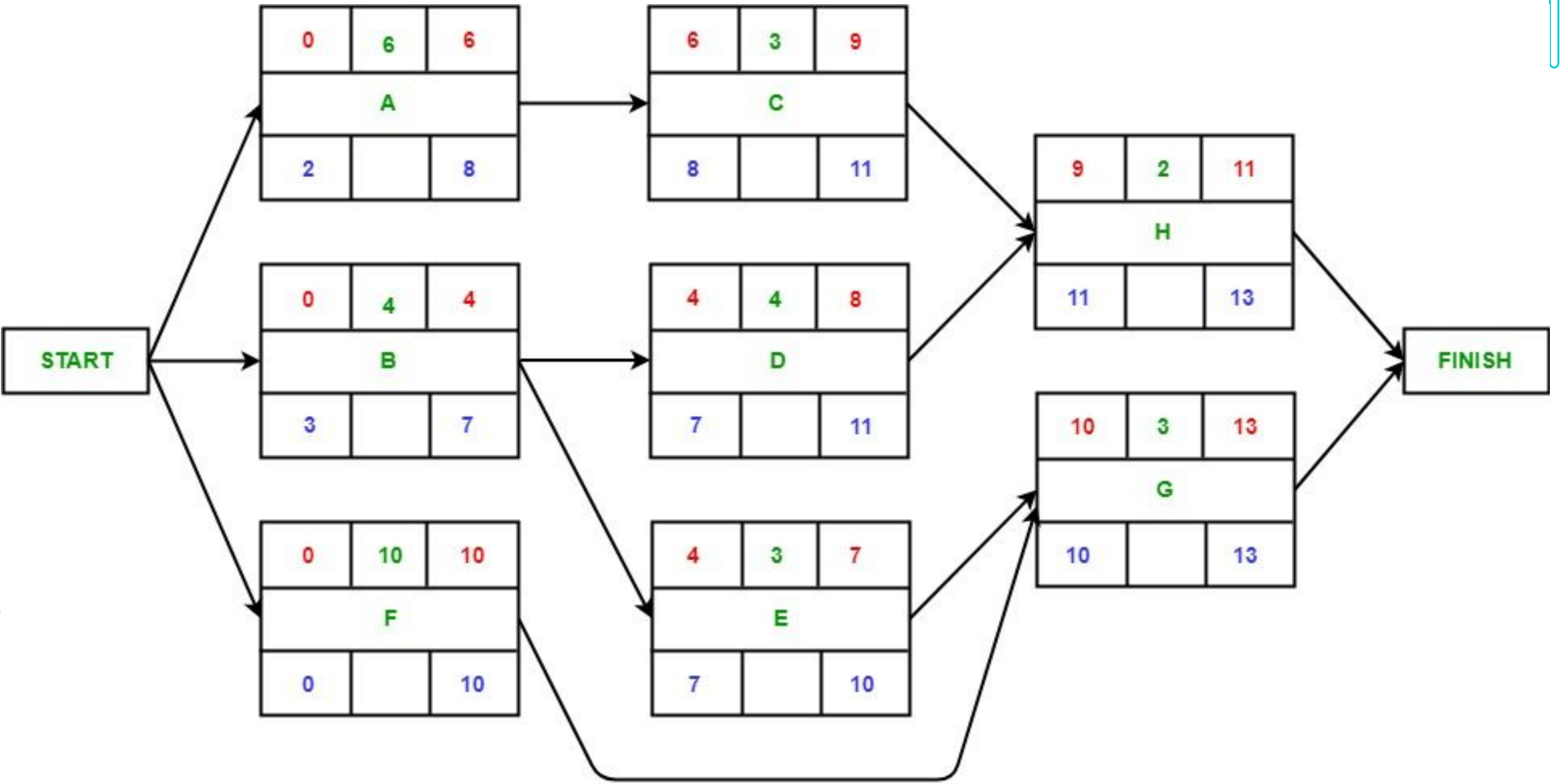
Forward Pass: After Calculation



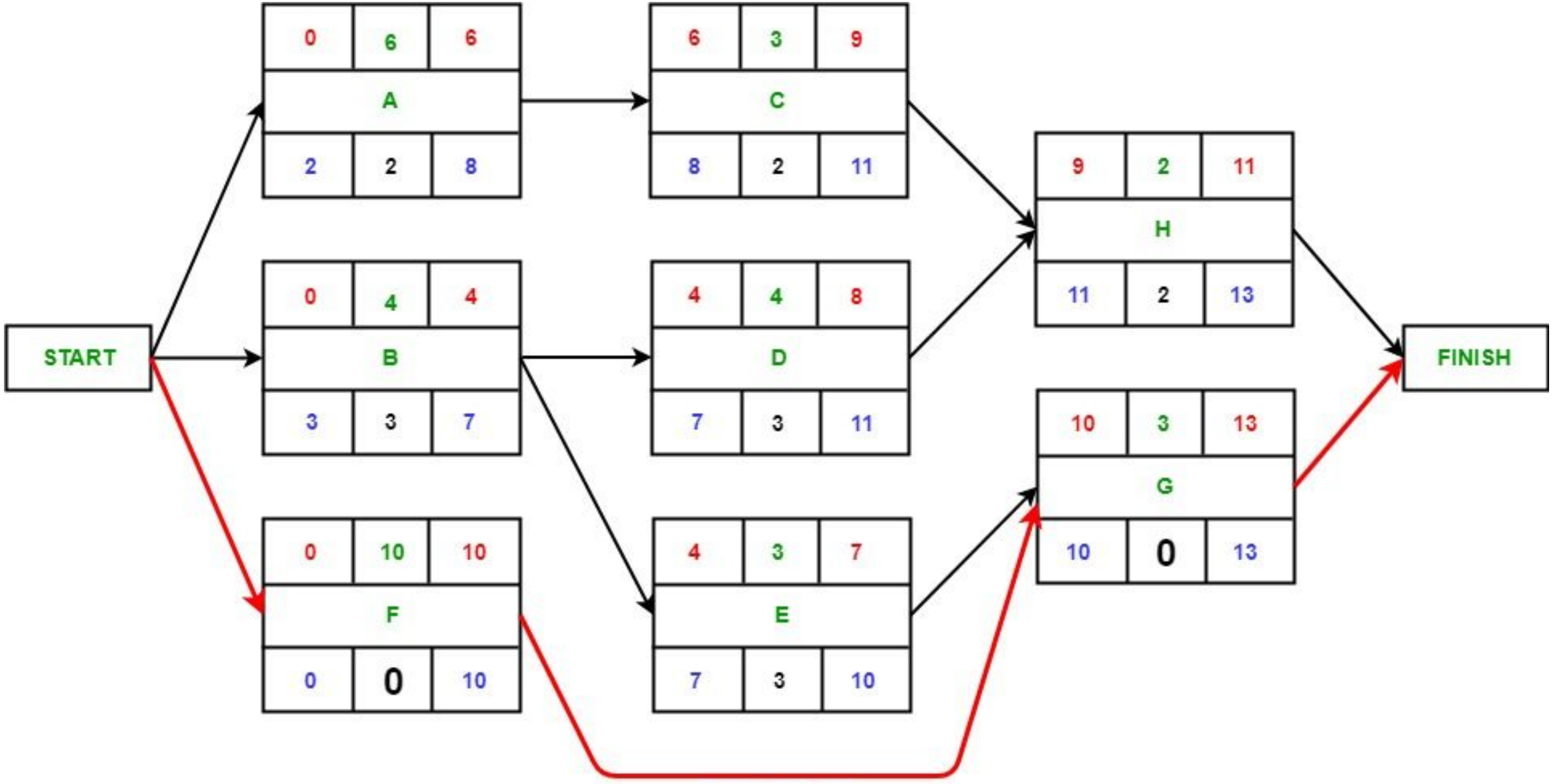
CPM-Backward Pass

- The **second stage** of the analysis of a critical path network is to carry out a backward pass to calculate the **latest date** at which each activity may be started and finished without delaying the end date of the project.
- For this, **we assume** that the latest finish date for the project is **as same** as the earliest finish date.
- **The backward pass rule:**
  - **Latest finish** date of a activity is the **latest start** date for an activity that commences immediately after that activity is complete.
  - Where **more than one activity** commence we take the **earliest** of the **latest start** dates for those activities.

Backward Pass: After Calculation



Identifying Critical Path: After Calculation



Critical Path

2.3.2

PERT:

PERT:

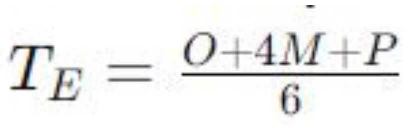
- A project management tool used to analyze the time required to complete tasks and identify the **probability** of meeting deadlines.
- **History:**
  - Developed in the 1950s by the **U.S. Navy** for the Polaris submarine missile project.
  - Originated as a method to manage large-scale, complex projects.
- **Why It's Used?**
  - To manage projects with **uncertain activity durations.**
  - Focuses on **improving accuracy** by using three-time estimates (***Optimistic, Pessimistic, and Most Likely***).

PERT:

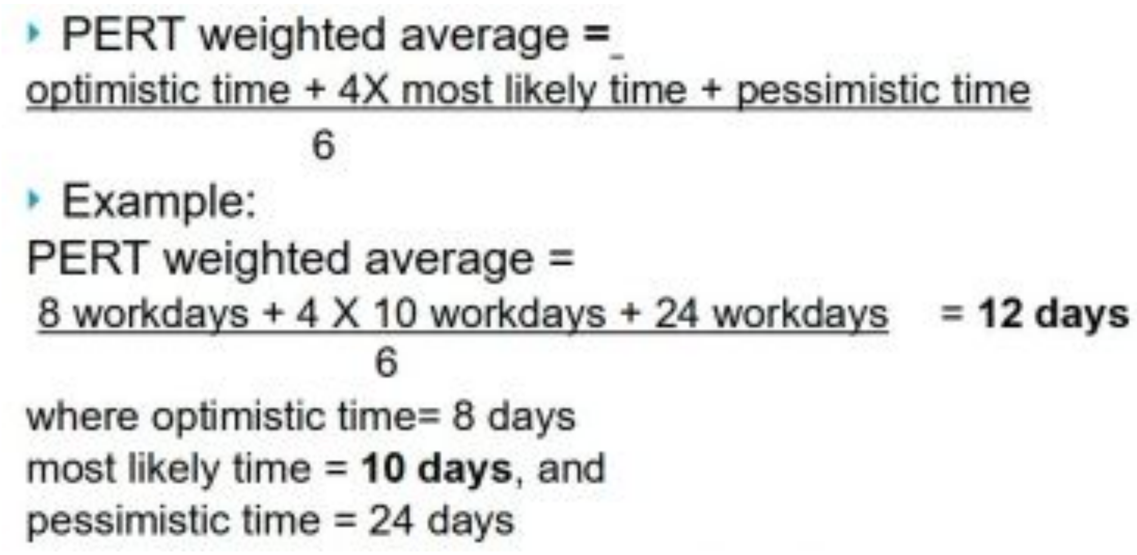
- Takes ***account of the uncertainty*** surrounding estimates of task durations.
- This method is similar to CPM technique but, **instead of** using a single estimate for the duration of each task, **PERT requires three estimates:**
  - **Most Likely Time (M):** The time we would expect the task **under normal circumstances.** Assumes standard delays or challenges.
  - **Optimistic Time (O or A):** The **shortest time** in which we would expect to complete the activity**. Assumes** no delays or obstacles.
  - **Pessimistic Time (P):** The **longest time** an activity might take if significant delays or issues occur. Represents the "**worst-case scenario.**"
- PERT is **calculated backward from a fixed end date** since deadlined typically cannot be moved.

PERT Expected Time (Te) Calculation:

❖ PERT uses these estimates to calculate the **expected time (Te)** for an activity:



$$T_E = \frac{O + 4M + P}{6}$$



- PERT weighted average =

$$\frac{\text{optimistic time} + \text{4X most likely time} + \text{pessimistic time}}{6}$$

- Example:

PERT weighted average =

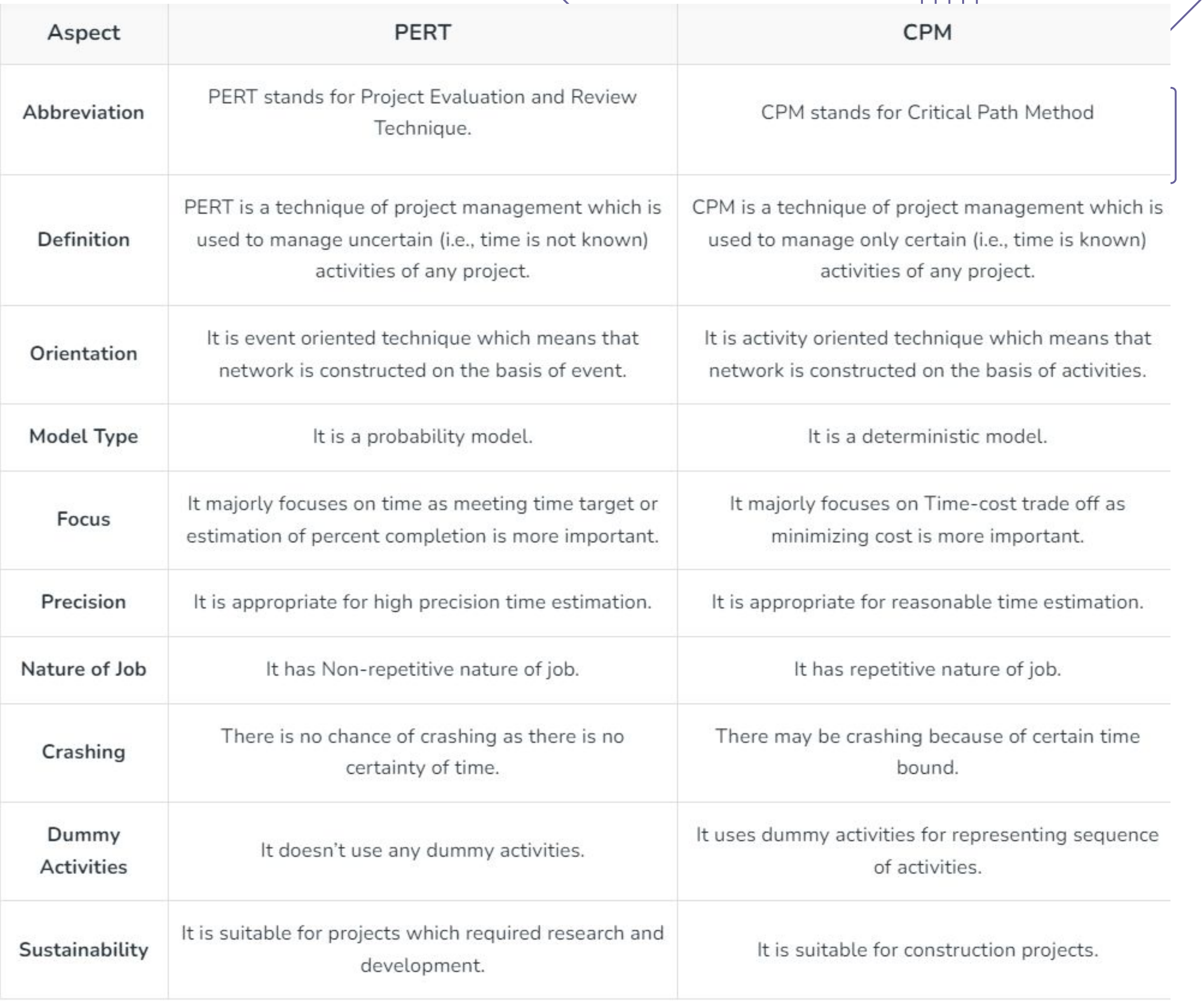
$$\frac{\text{8 workdays} + \text{4 X 10 workdays} + \text{24 workdays}}{6} = \textbf{12 days}$$

where optimistic time= 8 days
most likely time = **10 days**, and
pessimistic time = 24 days

Need / Benefits of a PERT:

- **Improved Project Planning:** PERT helps identify and define all tasks, dependencies, and milestones, creating a clear project roadmap.
- **Effective Time Management:** By using three-time estimates (Optimistic, Pessimistic, and Most Likely), it accounts for uncertainties and provides a realistic project schedule.
- **Risk Analysis and Management:** Helps identify potential bottlenecks and critical tasks, allowing proactive risk mitigation and contingency planning.
- **Enhanced Decision-Making:** Provides insights into task prioritization and resource allocation by identifying the critical path and tasks with slack.
- **Better Resource Allocation:** Helps optimize resource utilization by understanding task dependencies and timelines.
- **Flexibility and Adaptability:**Allows adjustments to schedules and plans based on changing project conditions and unforeseen delays.

PERT vs CPM:



Aspect	PERT	CPM
Abbreviation	PERT stands for Project Evaluation and Review Technique.	CPM stands for Critical Path Method
Definition	PERT is a technique of project management which is used to manage uncertain (i.e., time is not known) activities of any project.	CPM is a technique of project management which is used to manage only certain (i.e., time is known) activities of any project.
Orientation	It is event oriented technique which means that network is constructed on the basis of event.	It is activity oriented technique which means that network is constructed on the basis of activities.
Model Type	It is a probability model.	It is a deterministic model.
Focus	It majorly focuses on time as meeting time target or estimation of percent completion is more important.	It majorly focuses on Time-cost trade off as minimizing cost is more important.
Precision	It is appropriate for high precision time estimation.	It is appropriate for reasonable time estimation.
Nature of Job	It has Non-repetitive nature of job.	It has repetitive nature of job.
Crashing	There is no chance of crashing as there is no certainty of time.	There may be crashing because of certain time bound.
Dummy Activities	It doesn't use any dummy activities.	It uses dummy activities for representing sequence of activities.
Sustainability	It is suitable for projects which required research and development.	It is suitable for construction projects.

Shortening Project Duration

2.4

Shortening Project Duration:

Shortening the Project Duration **involves** reducing the time required to complete a project **without compromising quality or scope.**

➔ If we wish to shorten the overall duration of a project, we would normally consider attempting to **reduce activity durations.**

Below are the key approaches:

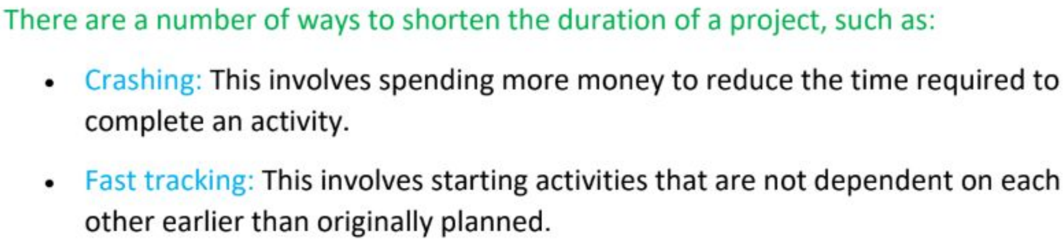
- **Focus on Critical Path Activities**
  - Identify activities on the critical path, as shortening these directly reduces project duration.
  - Avoid spending effort on non-critical activities, as they do not impact the overall timeline.
- **Add Resources**
  - Apply additional resources to critical tasks (e.g., hire more staff, increase working hours).
  - Be cautious to avoid inefficiencies due to overstaffing or resource constraints.

Shortening Project Duration:

- **Work Overtime**
  - Extend working hours or introduce shifts to speed up task completion.
  - Ensure overtime does not lead to burnout or reduced productivity.
- **Reallocate Tasks**
  - Adjust resource allocation by prioritizing critical tasks over non-critical ones.
  - Transfer resources from less critical tasks to critical path activities.
- **Parallel Execution**
  - Increase parallelism by performing tasks simultaneously instead of sequentially.
  - Review dependencies to determine which tasks can overlap or start earlier.

Shortening Project Duration:

- **Revisit Activity Durations**
  - Reassess and optimize task durations without compromising the deliverable quality.
  - Explore improved methods or tools to perform tasks more efficiently.
- **Review Bottlenecks**
  - Identify and remove delays caused by resource, equipment, or process bottlenecks.
- **Adjust Work Methods**
  - Reevaluate and improve the sequencing or methods used to perform tasks.



There are a number of ways to shorten the duration of a project, such as:

- Crashing: This involves spending more money to reduce the time required to complete an activity.
- Fast tracking: This involves starting activities that are not dependent on each other earlier than originally planned.

Identifying Critical Activities

2.5

Identifying Critical Activities:

Identifying critical activities is **crucial for ensuring timely project completion** and preventing **potential delays.** These activities, also known as "**critical path tasks**," directly impact the overall project duration and must be completed on schedule to stay within the established timeframe.

- **What are critical activities?**

  Definition:

  - Critical activities are tasks within a project's network that have **zero slack time**. This means that any delay in their completion directly translates to a **delay in the entire project deadline.**

- **Critical Path:**

  - These activities lie on the critical path, which is the **longest sequence of dependent tasks** that must be finished sequentially to achieve project completion.

Why is it important to identify them?

Knowing your critical activities allows you to:

- **Focus resources:**
  - Prioritize resources (personnel, budget, tools) towards accomplishing critical tasks first.
- **Monitor closely:**
  - **Proactively track progress** of critical activities to avoid potential bottlenecks and take corrective action if needed.
- **Manage risks:**
  - Identify and mitigate potential risks that could affect critical activities and derail the project schedule.
- **Make informed decisions:**
  - Understand the impact of potential delays or scope changes on the critical path and adjust accordingly.

How to identify critical activities?

Several methods can help us pinpoint critical activities in software projects:

❖ **Critical Path Method (CPM):** This traditional technique involves creating a network diagram of project tasks and their dependencies, calculating task durations, and then using calculations to identify the critical path.

❖ **PERT (Program Evaluation and Review Technique):** This method incorporates uncertainty in task durations by using probabilistic estimates, providing a more nuanced view of the critical path and potential risks.

❖ **Project management software:** Many project management tools offer built-in features for network analysis and critical path identification, simplifying the process.

Self Study Topics:

1. Why is planning necessary? Highlight on the steps of activity planning.
2. WBS vs PBS
3. Significance of Float
4. Total Float and Free Float

PYQs:

1. What is forward pass & backward pass? Explain 2021 5 Marks

THANKS!

Do you have any questions?

theciceerguy@p m.me

9841893035

ghimires.com.np