4COSC002W Mathematics for Computing

Lecture 5

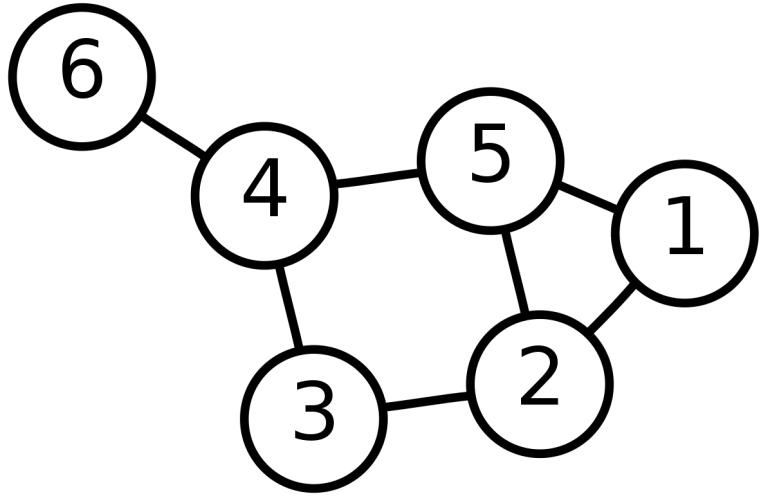
Graph Theory

UNIVERSITY OF WESTMINSTER

Introduction to Graph Theory

Graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects from a certain collection.

A graph comprises vertices (or nodes) and edges (or links) that connect these vertices.



Applications of Graphs and Trees in computing disciplines

- Network
- Graph Databases
- Circuits
- Modelling – state diagrams, flow diagrams
- Search
- AI applications
- Process optimisation – Logistics, Project management

Königsberg Bridge Problem

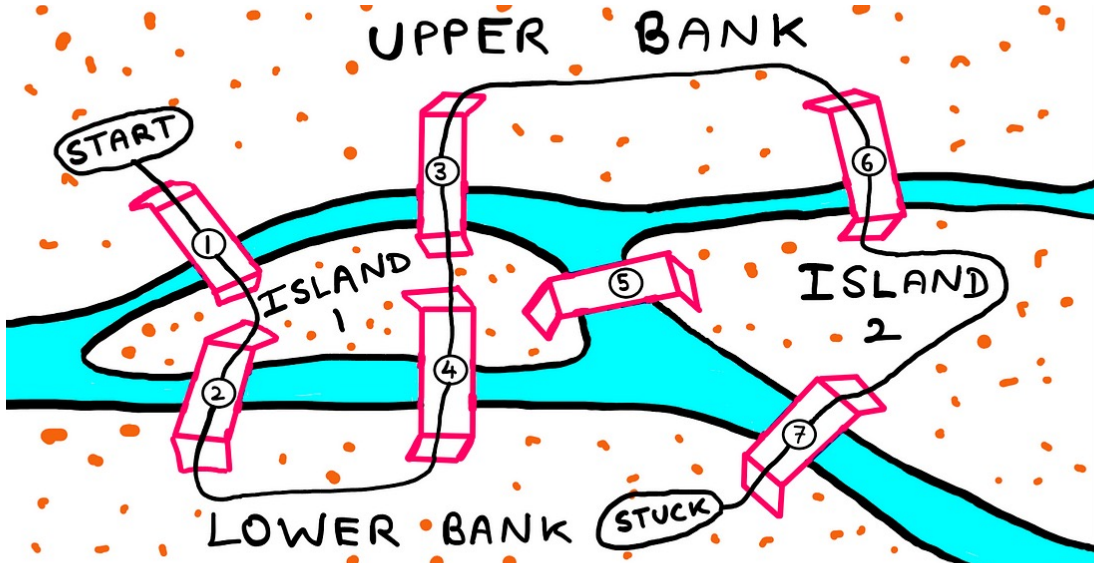
Problem Statement:

Can one walk through the city of Königsberg cross each of its seven bridges once and only once and return to the starting point?

Leonhard Euler (1736):

Reimagined the problem as one involving nodes and links, leading to the formation of early *graph theory*.

Proved it's impossible by introducing the concept of 'degree of a vertex'.



The city of Königsberg in Prussia (now Kaliningrad, Russia) was built on both sides of the Pregel River, including two large islands connected to each other and the mainland by seven bridges.

Directed vs Undirected Graphs

If the directions of the edges matter, then we show the edge directions, and the graph is called a *directed graph* (or a *digraph*)

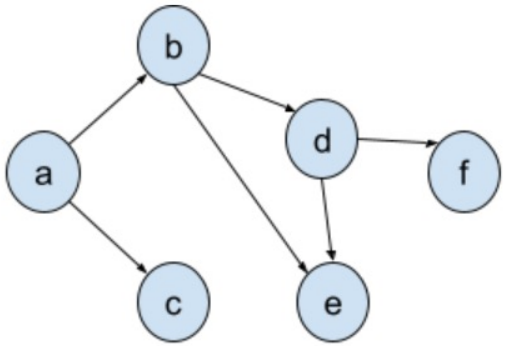
If the direction of the edges does not matter, the graph is called an *undirected graph*.

Graph Cyclicity

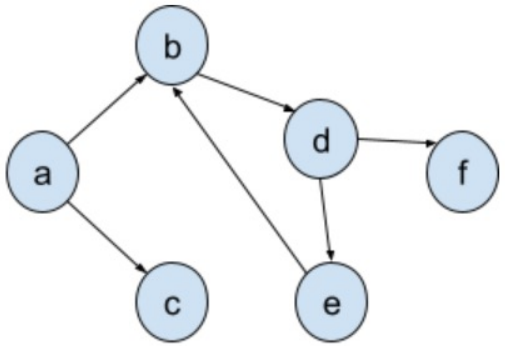
A cycle in a graph G is a path where the last node is the same as the first node.

A graph is *cyclic* if it has at least one cycle. Otherwise, it is *acyclic*.

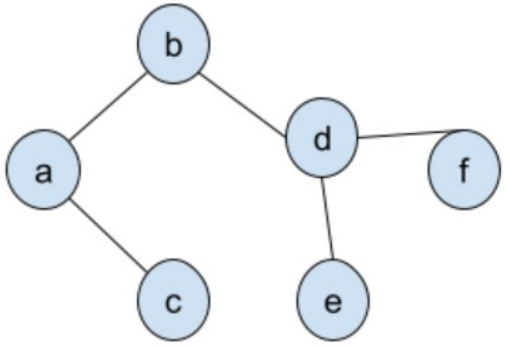
Examples of Graphs



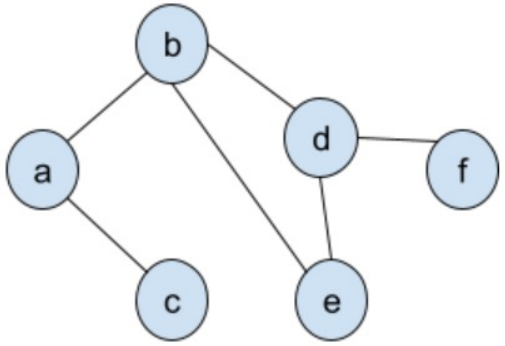
Acyclic (Directed) Graph



Cyclic (Directed) Graph



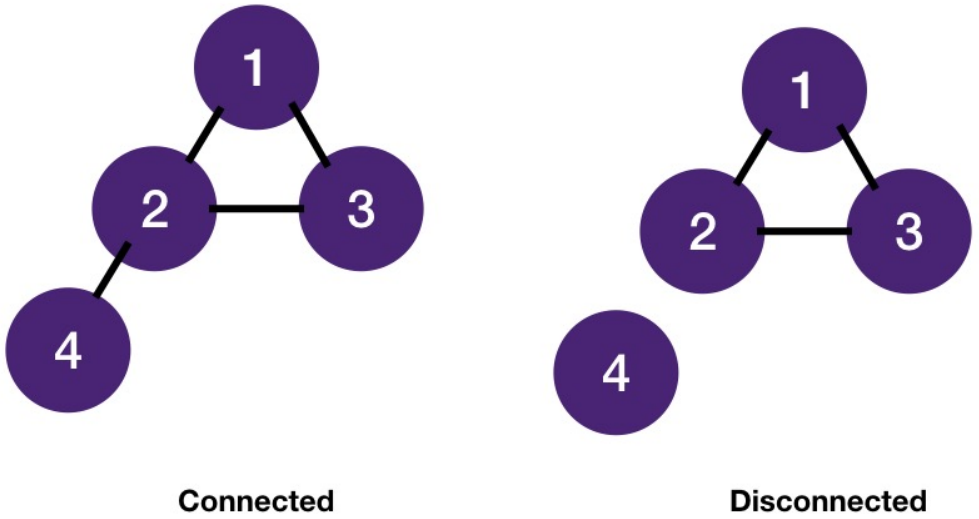
Acyclic (Undirected) Graph



Cyclic (Undirected) Graph

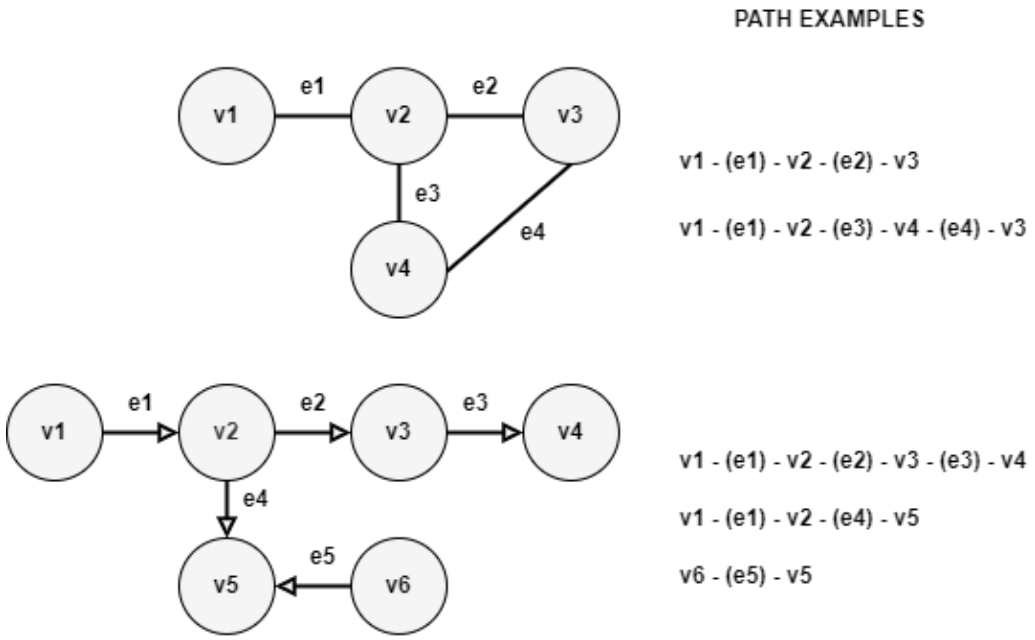
Graph Connectivity

- A graph is *connected* if for every pair of nodes, there is a path between them. Otherwise, the graph is *disconnected*.



Path in a Graph

A path in graph G is a sequence of nodes $x_1, x_2, \ldots, x_k$, such that for any node x_i $(1 \leq i \leq k\text{-}1)$, there is an edge from it to the next one in the sequence.



Graph Notation

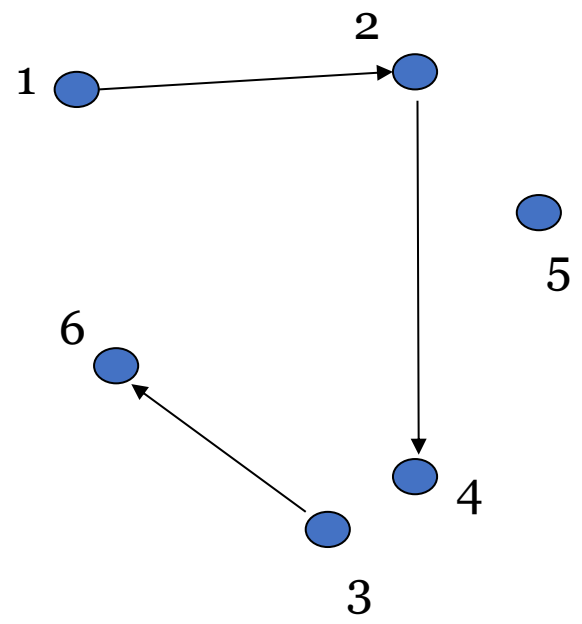
Formally, a graph G is a structure (V, E) where

- V is a finite set of nodes, and
- E that is a set of directed or undirected pairs of nodes x and y from V

If a pair of nodes x and y is *directed*, we abbreviate it as (x,y)

If a pair of nodes x and y is *undirected*, we abbreviate it as $\{x,y\}$

Example: Graph Notation



V={1,2,3,4,5,6}

E={(1,2), (2,4), (3,6)}

This is an *acyclic*, *directed*, *disconnected* graph.

Note the directions!

There is some pattern represented here - E is a set of all pairs (x,y) such that $y = 2x$

Basic Graph Traversals

Graph traversal is a technique used to visit all the vertices of a graph in a particular sequence. Traversing a graph means examining all its vertices and edges. Two common ways to traverse are Depth-First Search (DFS) and Breadth-First Search (BFS).

Depth-First Search (DFS): A strategy that dives deep into a graph, visiting a vertex and then recursively visiting all its adjacent vertices before backtracking.

Applications: Finding connected components in a graph, Topological sorting, Pathfinding in maze puzzles, Checking for cycles in graphs.

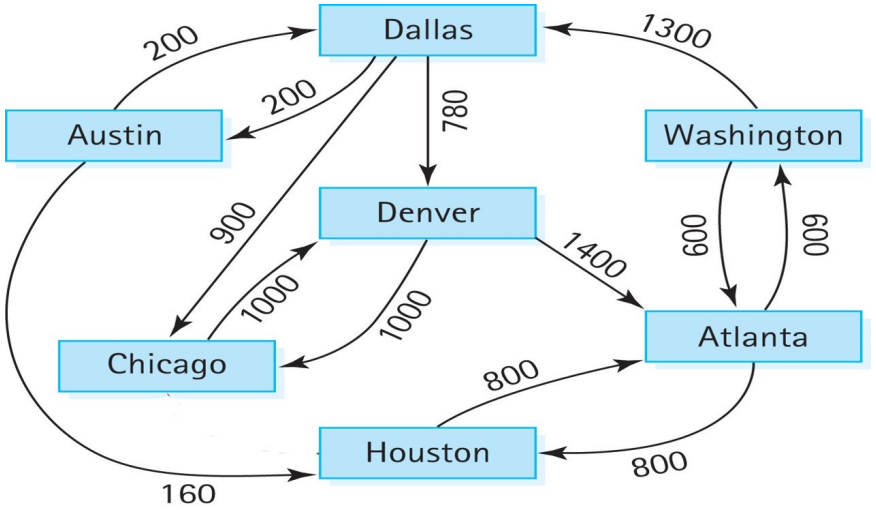
Breadth-First Search (BFS): A layer-wise traversal method, visiting a vertex and all its immediate neighbours before proceeding to their neighbours.

Applications: Shortest path in unweighted graphs, Finding connected components in undirected graphs, Broadcasting in a network, Crawling websites – web spiders use BFS to index pages.

Weighted Graphs

Weighted graphs are graphs in which each edge has an associated numerical value, often called weight.

This weight can represent concepts like distance, cost, etc., depending on the application.



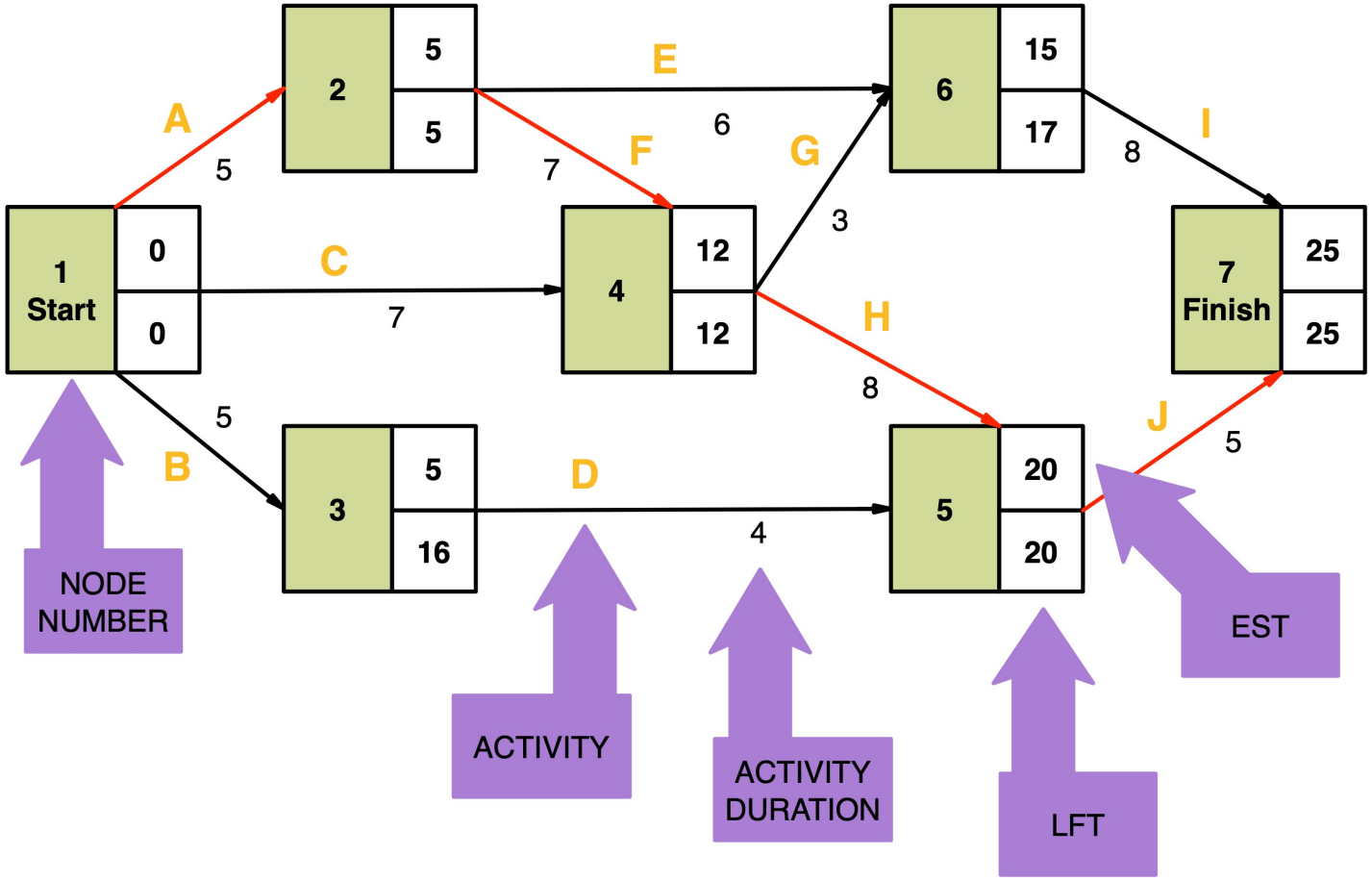
Critical Path Method

The Critical Path Method (CPM) is used in project management to determine the longest path in a schedule, which helps identify the *shortest possible project duration*.

It's often used with PERT (Program Evaluation Review Technique).

Example Graph:

Critical Path is 1, 2, 4, 5, 7



Critical Path Calculations

Forward Pass – to determine the earliest start (ES) and earliest finish (EF) for each activity:

- **ES** for the first task = 0.
- **EF** for an activity = ES + Activity duration.
- **ES** for subsequent tasks = Maximum EF of all *predecessor* tasks.

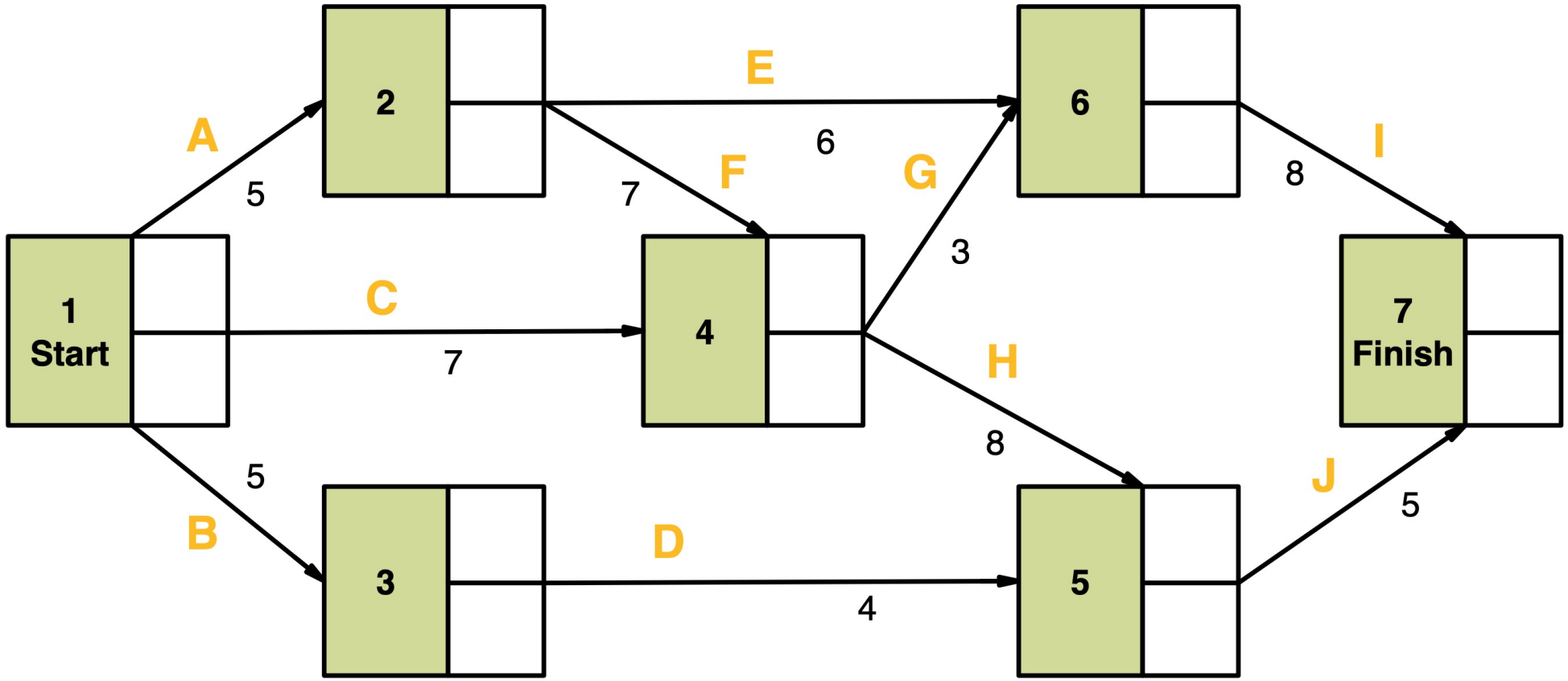
Backward Pass – to determine the latest start (LS) and latest finish (LF) for each activity:

- **LF** for the last task = EF of the last task in the forward pass.
- **LS** for an activity = LF – Activity duration.
- **LF** for a task with successors = Minimum LS of all *successor* tasks.

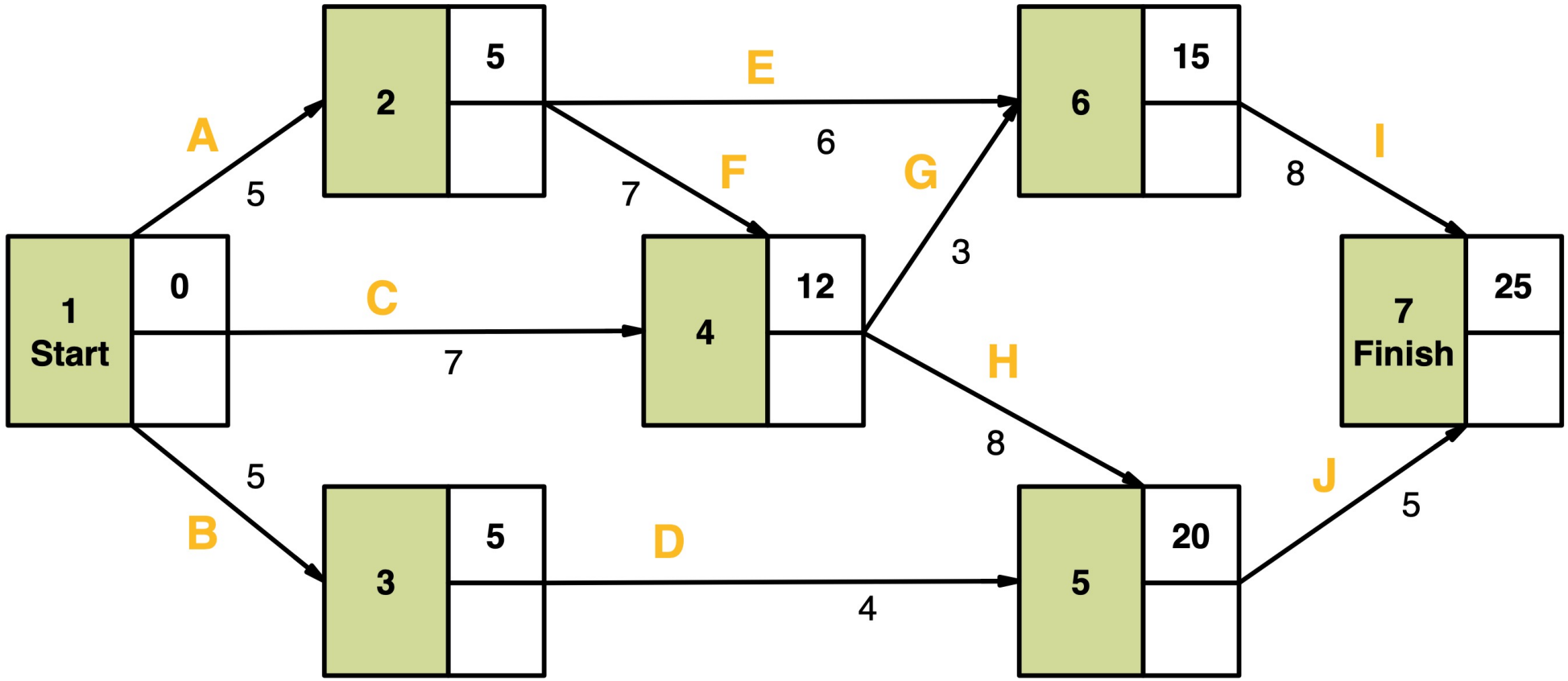
Slack/Float – how long an activity can be delayed without delaying the project.

- **Total Float** = LS - ES or LF - EF.

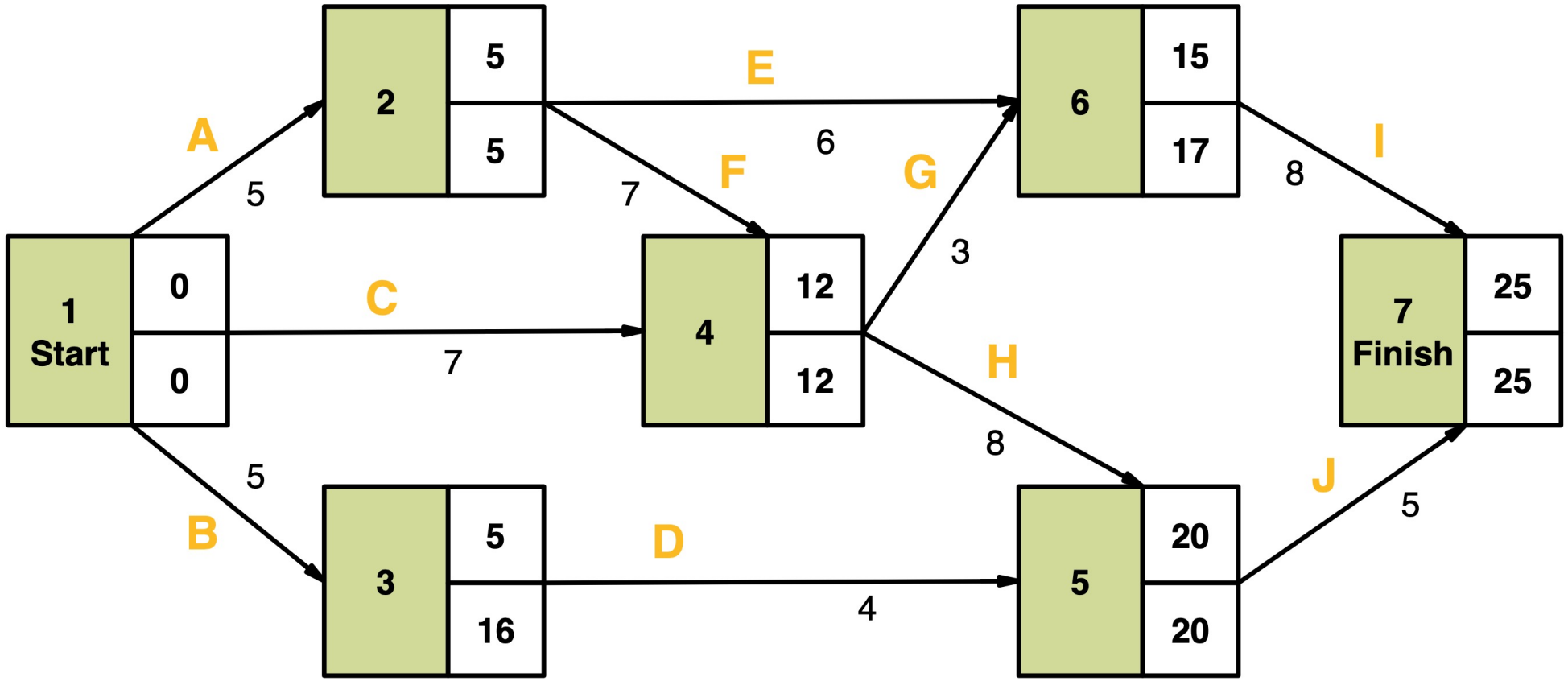
Example: Project Management using CPM



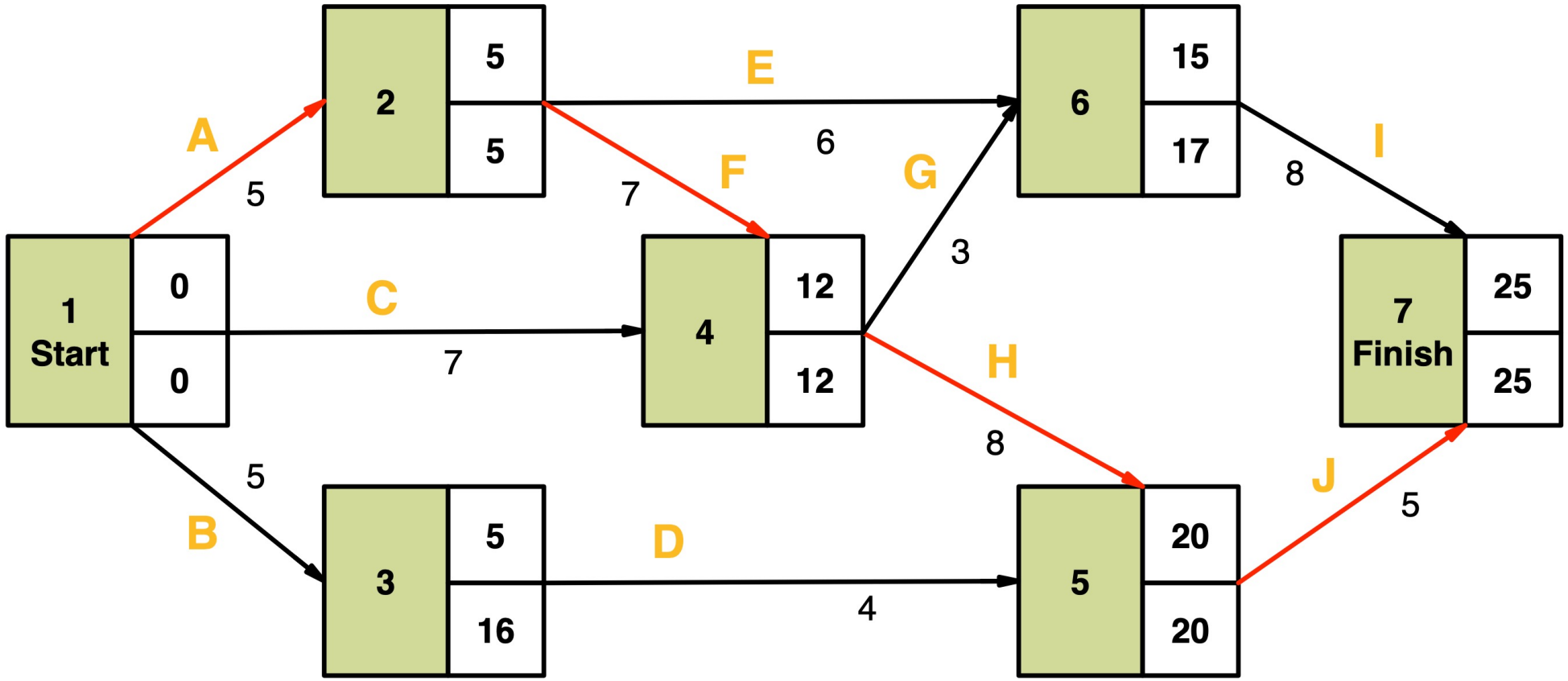
Example: Project Management using CPM



Example: Project Management using CPM



Example: Project Management using CPM



Application of Weighted Graphs and Critical Paths

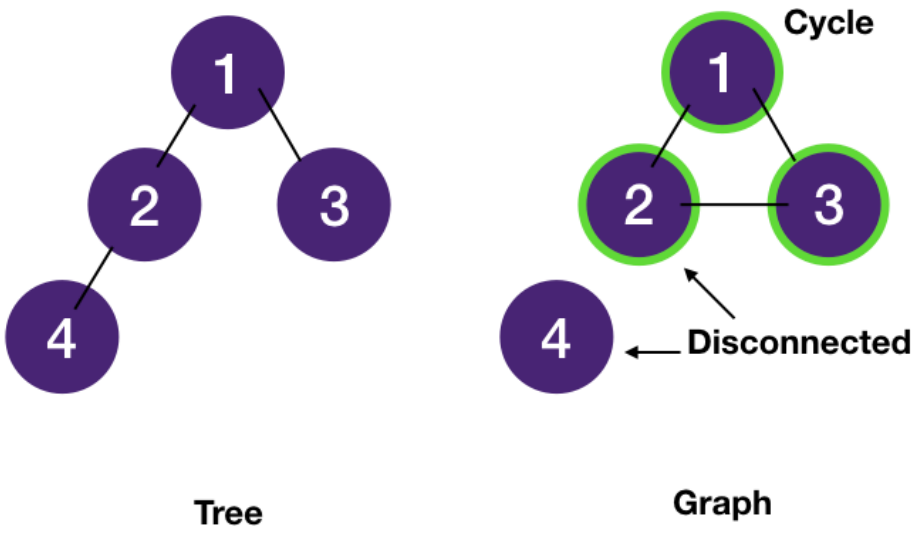
Weighted Graphs:

- **Transportation:** Represents roads with weights as distances or travel time between cities (nodes).
- **Telecommunication:** Represents connections (edges) between nodes (communication towers) with weights as latency or bandwidth.
- **Supply Chain:** Shows transport routes (edges) between production units, warehouses, and retailers (nodes) with weights as costs or delivery time.

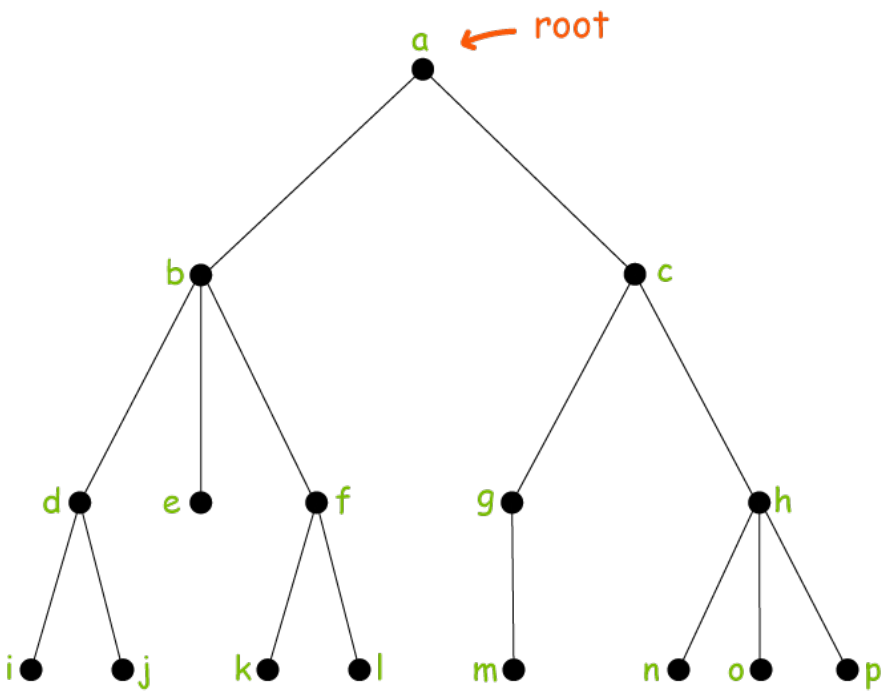
- **Critical Paths:**
  - **Project Management:** Determines the shortest time to complete a project, helping managers allocate resources efficiently and identify delay-sensitive activities.
  - **Manufacturing:** Streamlines production processes by identifying the most time-sensitive steps.
  - **Construction:** Determines the sequence and timing of tasks, ensuring timely completion of projects.
  - **Software Development:** Helps schedule and prioritise tasks, ensuring timely software releases.

Trees

- A tree is a connected acyclic undirected graph.
- In a tree, any two nodes are connected by *exactly one path*.



Tree-Related Concepts



*This tree has 9 leaves (i, j, e, k, l, m, n, o, p)

- The nodes adjacent to node x and below x are called the *children* of x, and x is called their *parents*
- A node that has no children is called a *leaf*
- The *descendants* of a node are its *children*, their *children*, all the way down
- The *ancestors* of a node are its *parent*, its *grandparent*, all the way to the root

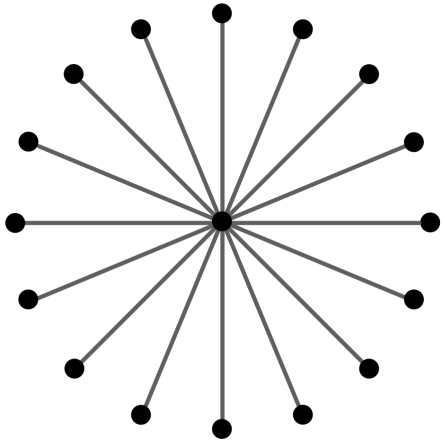
Tree Depth

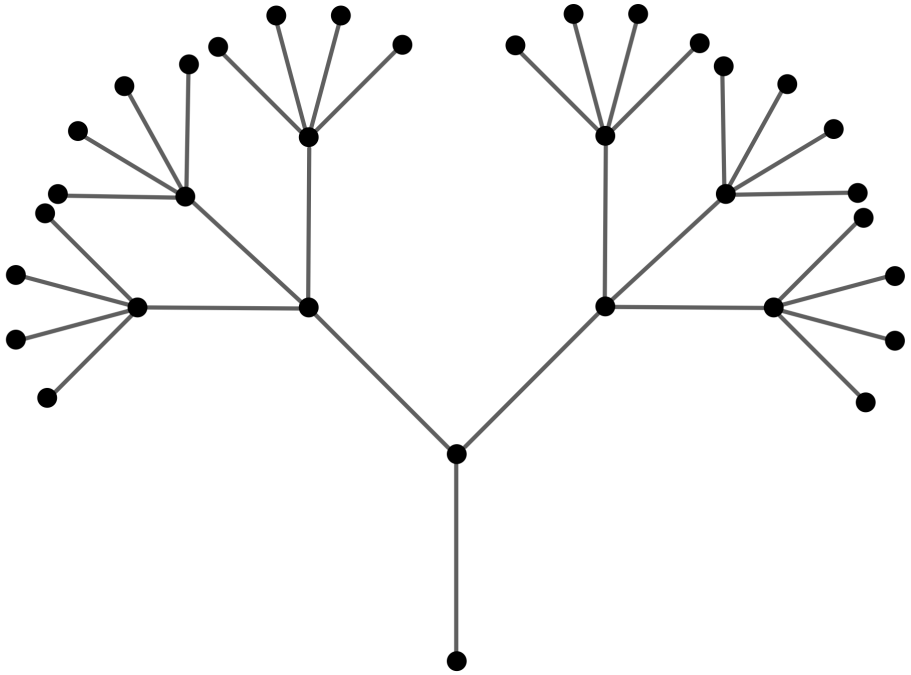
The *depth of a node* is the number of edges from the root to that node.

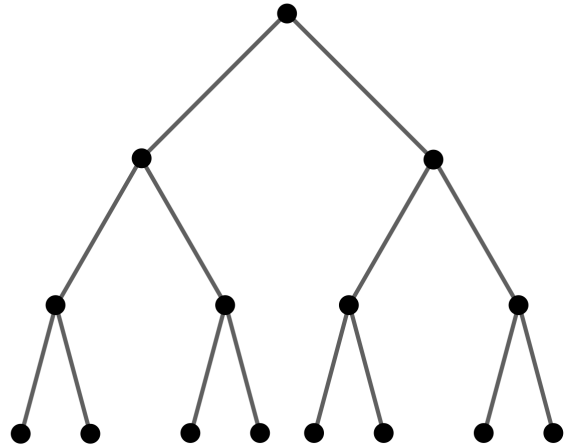
The *depth (*or *height)* of a rooted tree is the depth of the lowest leaf.

Activity: What is the depth of the tree from the above slide?

Trees Visual Representation

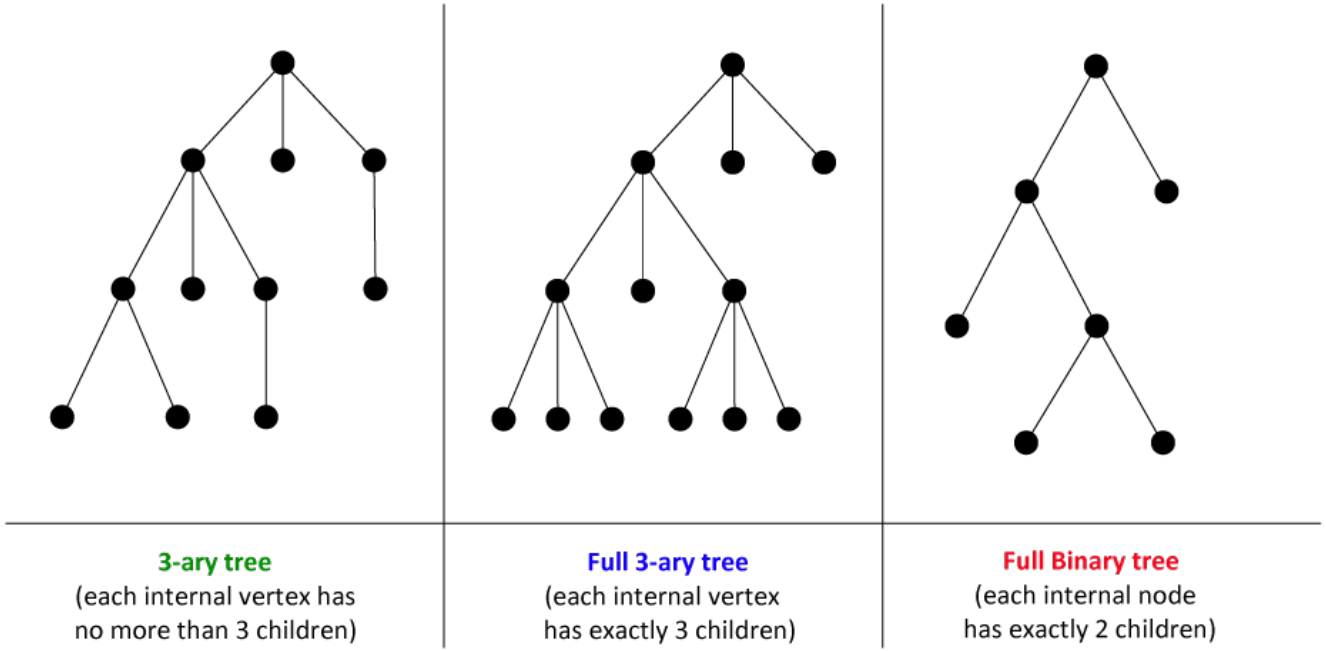




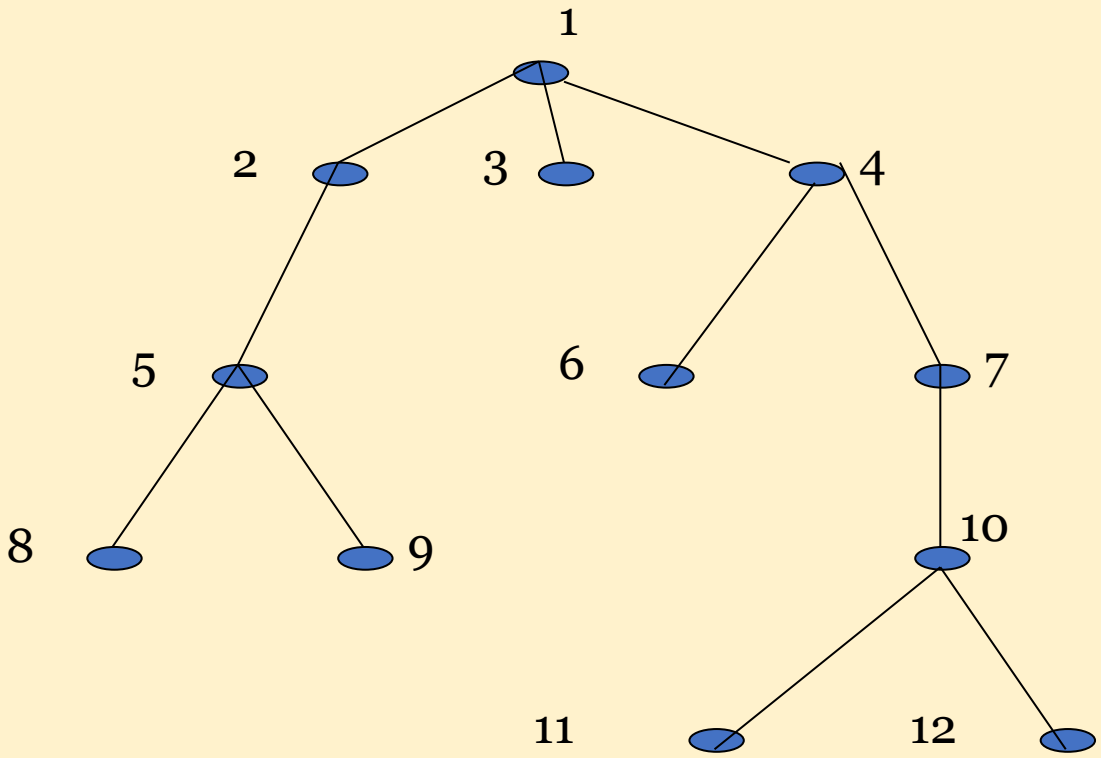


Binary Trees

A tree is *a binary tree* if every node has at most two children.



ACTIVITY: Name the features of the given tree

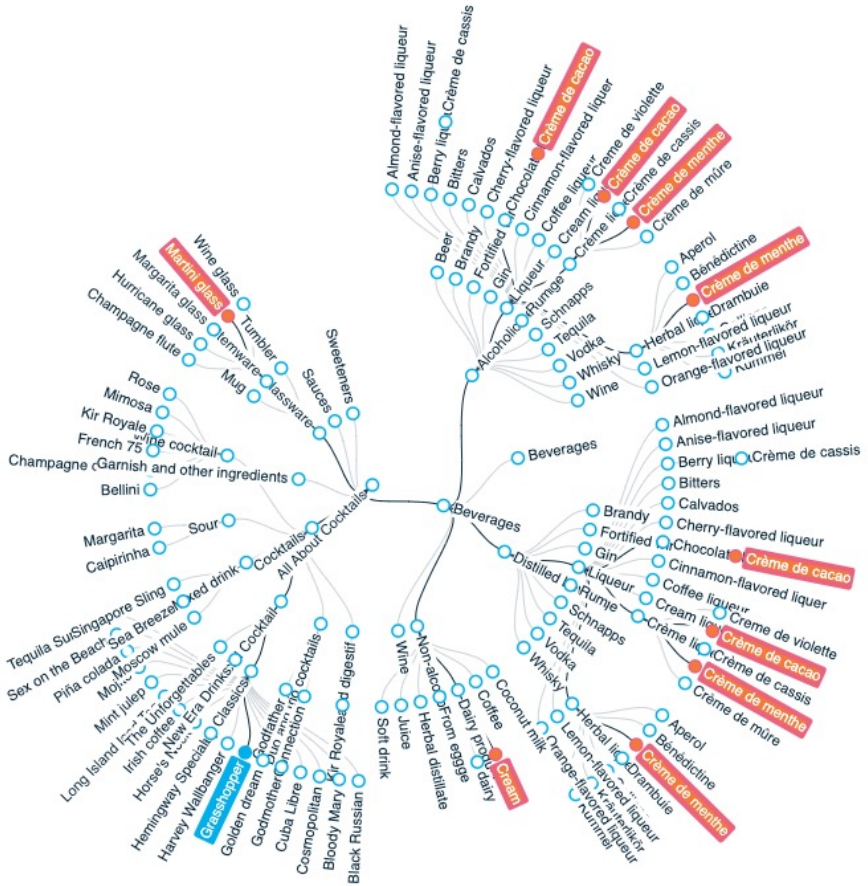


Graph Theory Real Life Application: Ontologies

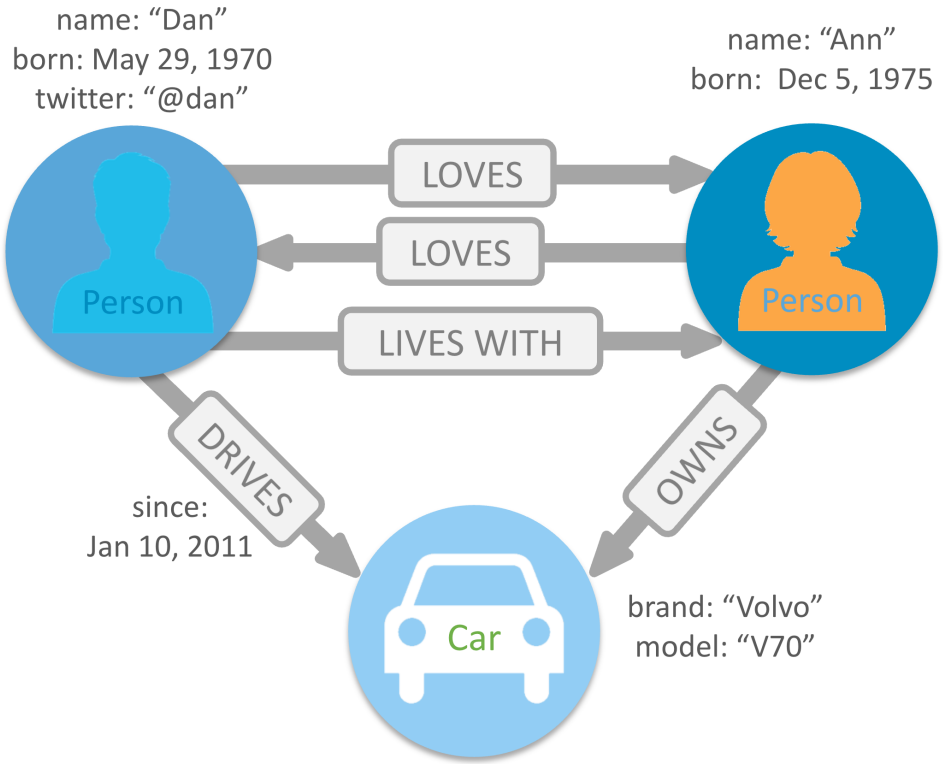
Ontologies define a set of concepts and the relationships between those concepts, typically within a specific domain.

They provide a shared and common understanding of a domain that can be communicated between people and systems.

Ontologies are naturally represented as *graphs*.



Graph Theory Real Life Application: Neo4j Database



Neo4j is a graph NoSQL database management system known for its native graph storage and processing.

Cypher Query Language: Used to interact with the Neo4j database.

Benefits of Neo4j:

- *Native Graph Processing:* Optimized for graph-related operations.
- *Rich Ecosystem:* Thriving community, plugins, and integrations.
- *Real-time Insights:* Delivers immediate data analysis.
- *Flexibility:* Adaptable to changing data structures.
- *Relationship Handling:* Excellently manages intricate relationships.

neo4j