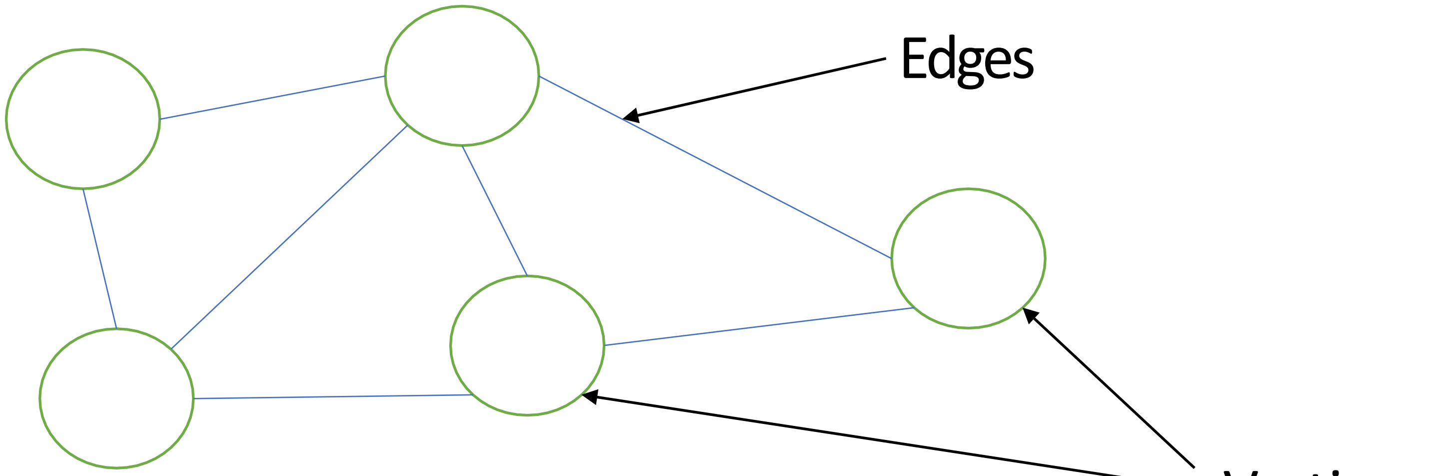
Data Structures and Analysis of Algorithms
CST 225-3

Introduction to Graphs

Uva Wellassa University

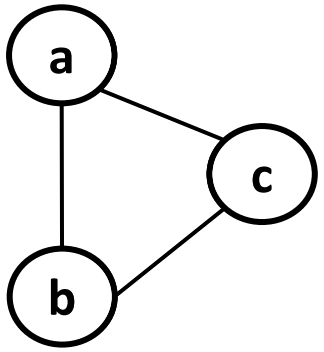
What is a Graph?

- A graph is a non-linear data structure that consists of vertices(or nodes) and edges which connect a pair of nodes.



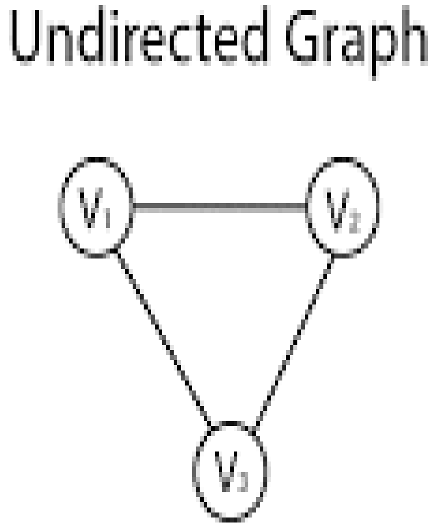
What is a Graph?

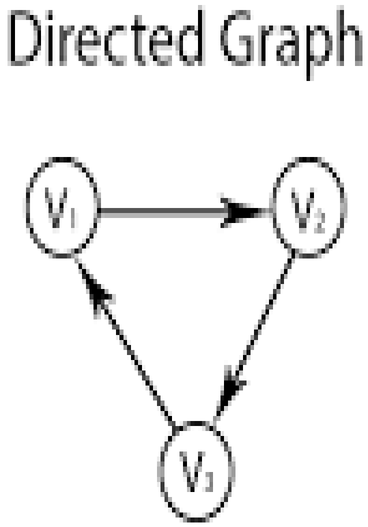
- Graph: a data structure containing
  - a set of vertices V
  - a set of edges E, where an edge represents a connection between 2 vertices
- The graph at right:
  - V = {a, b, c}
  - E = {(a, b), (b, c), (c, a)}



Directed vs. Undirected Graphs

- When the edges in a graph have a direction, the graph is called directed (or digraph).
- When the edges in a graph have no direction, the graph is called undirected.



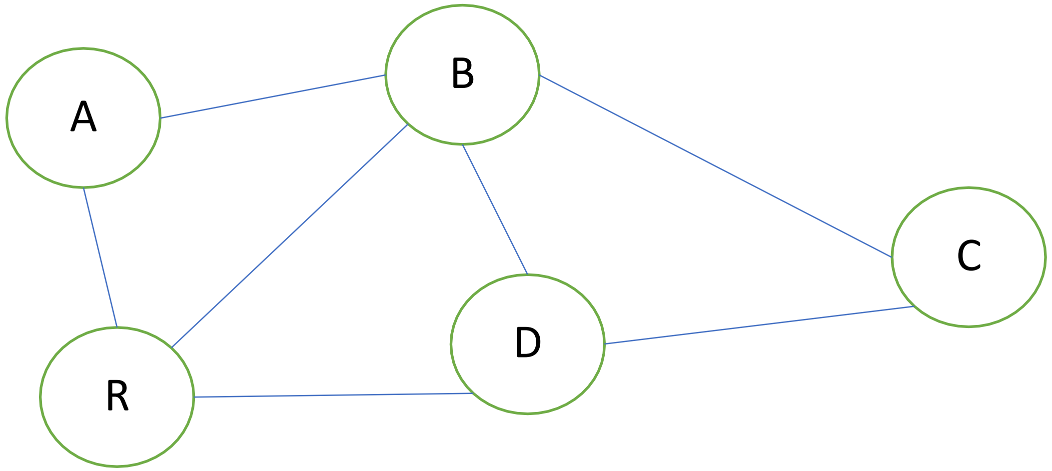


Graph Terminology

- The followings are some terminologies associated with graphs;
- **Vertex**– Each node of the graph is called as a vertex.
- **Edge**– Edge represents a path between two vertices or a line between two vertices.
- **Adjacency**– Two node or vertices are adjacent if they are connected to each other through an edge.
- **Path**– Path represents a sequence of edges between the two vertices.

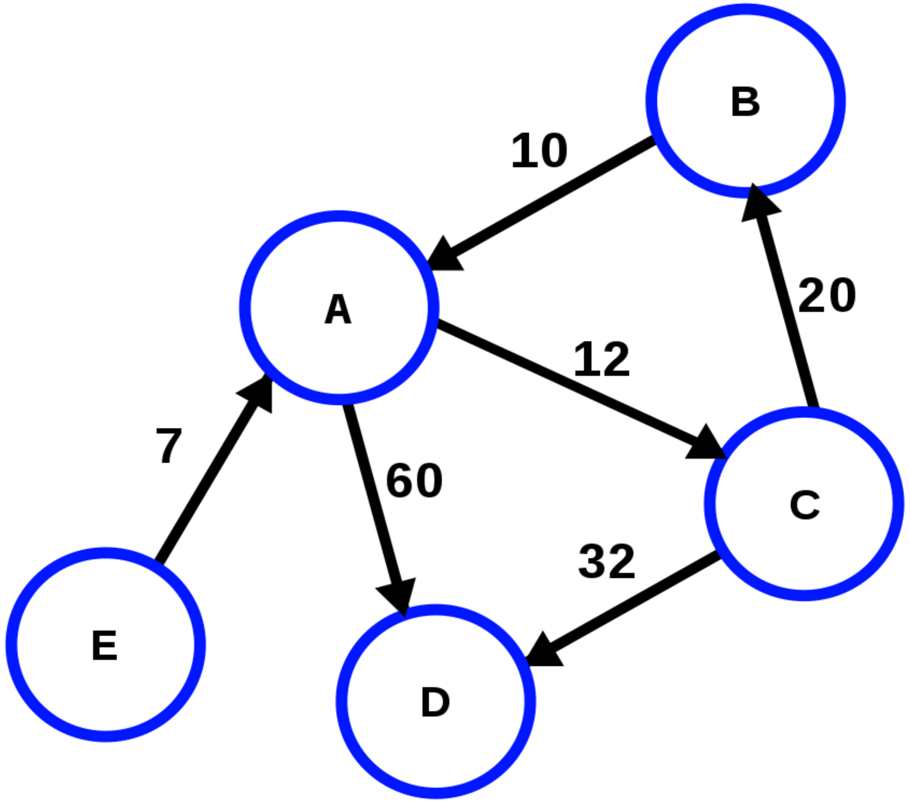
Graph Terminology cont'd

- Order of graph |V|= 5
- Size of graph |E|= 7



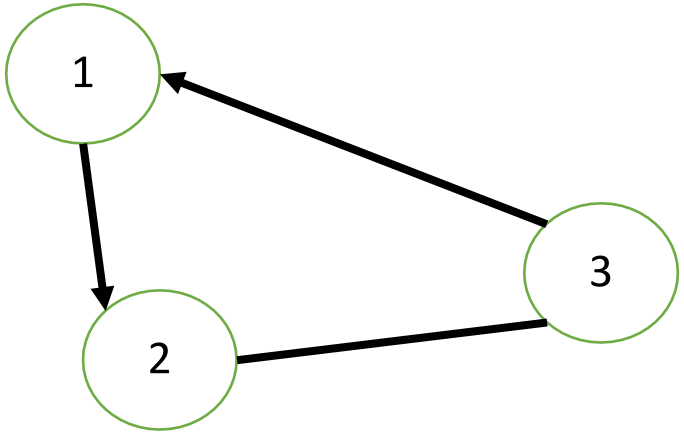
Weighted Graphs

- A weighted graph is a graph in which each branch is given a numerical weight.



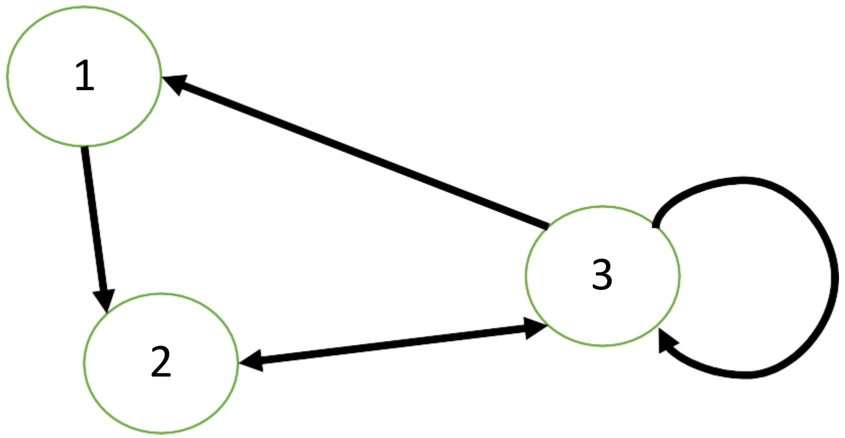
Mixed Graph

- A graph with both undirected and directed edges is said to be mixed graph.



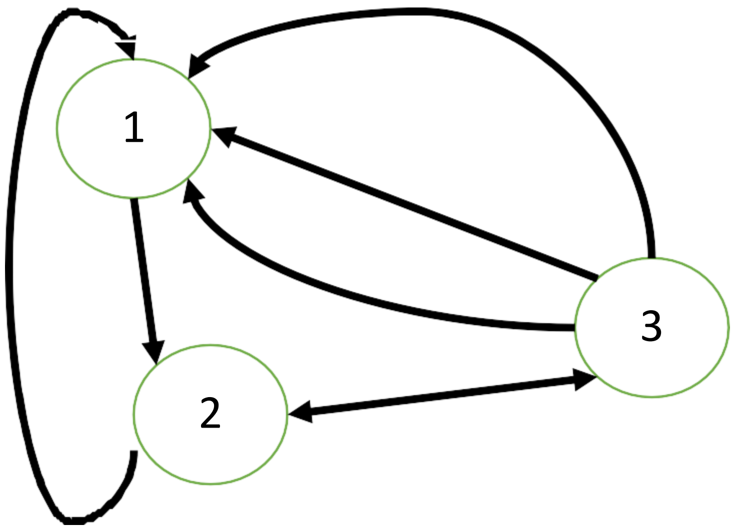
Graphs with Self Loop Edges

- Vertices can have edges directed to itself.



Graphs with Multi Edges

- Nodes can have many parallel edges between them.



- Graphs with no multi edges and self loops are **simple graphs** which can be directed or undirected.

Graph Representation

- There are two ways to represent graphs.
  1. Adjacency Matrix
  2. Adjacency List

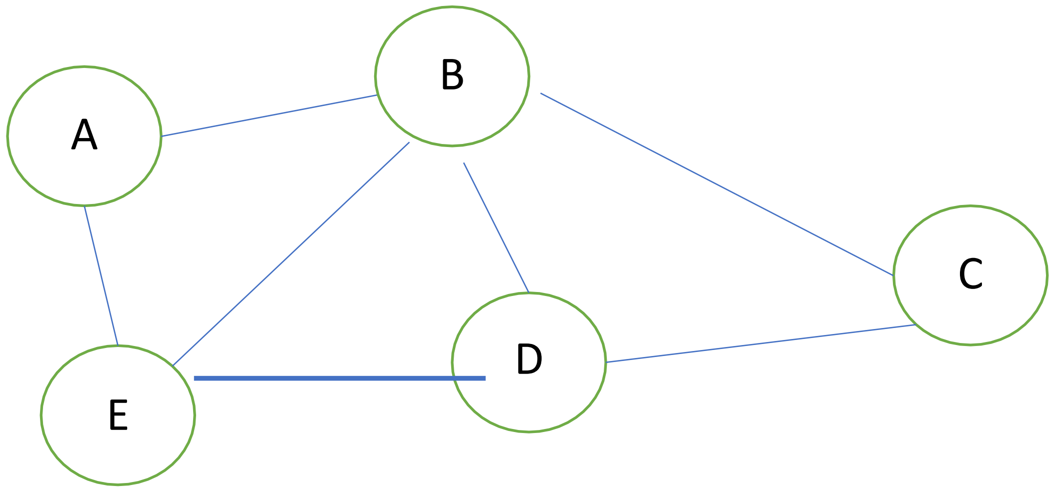
Adjacency Matrix

- Adjacency Matrix is a 2D array of vertices where each row and column represents a vertex.
- If a value of any element in the 2D array is 1, it represents that there is an edge connecting the vertices.

- The Adjacency Matrix A=(i,j) of a graph G=(V,E) with |V| nodes/vertices is an |V| X |V| matrix.
- A=[i][j] = 1 if i & j are adjacent

  = 0 otherwise

Adjacency Matrix - Undirected Graph



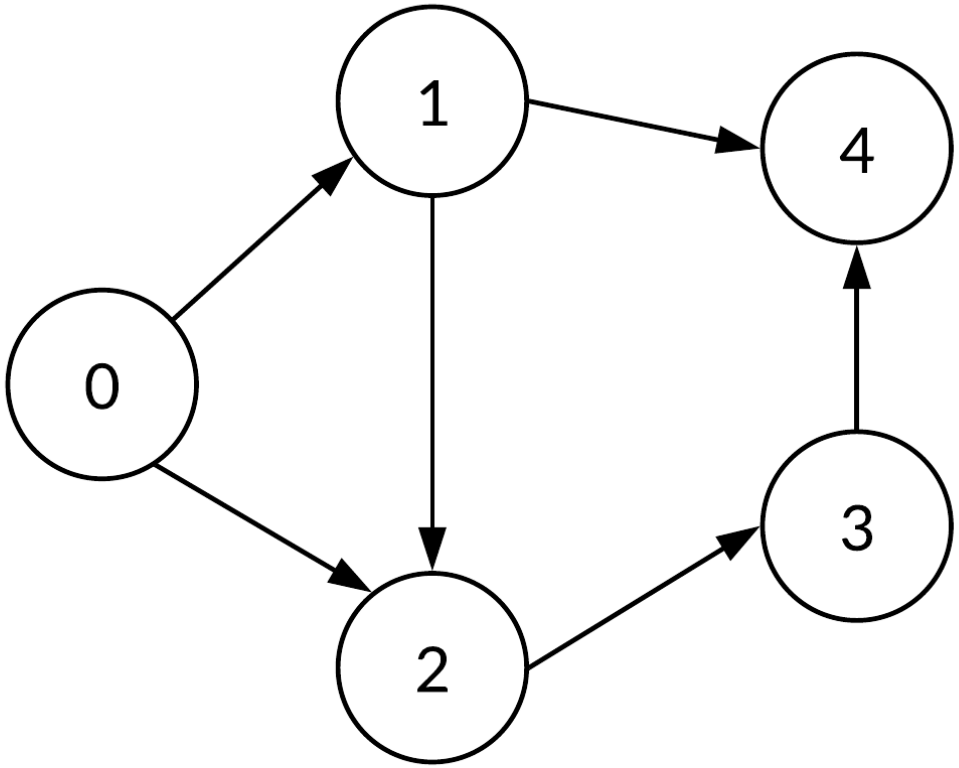
Index	Vertex Name
1	A
2	B
3	C
4	D
5	E

Vertex List

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Adjacency Matrix

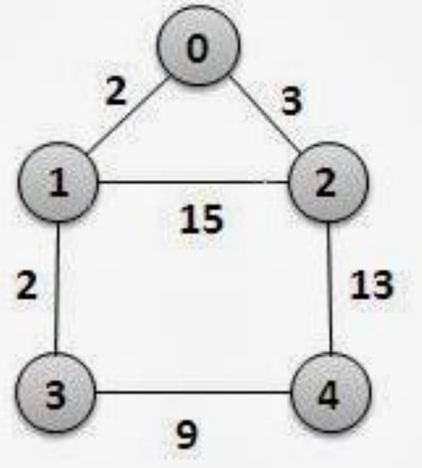
Adjacency Matrix - Directed Graph



	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Adjacency Matrix of a Weighted Graph

- The weight of the edge can be shown in the matrix when the vertices are adjacent.
- A nil value (0 or α) depending on the problem is used when they are not adjacent.

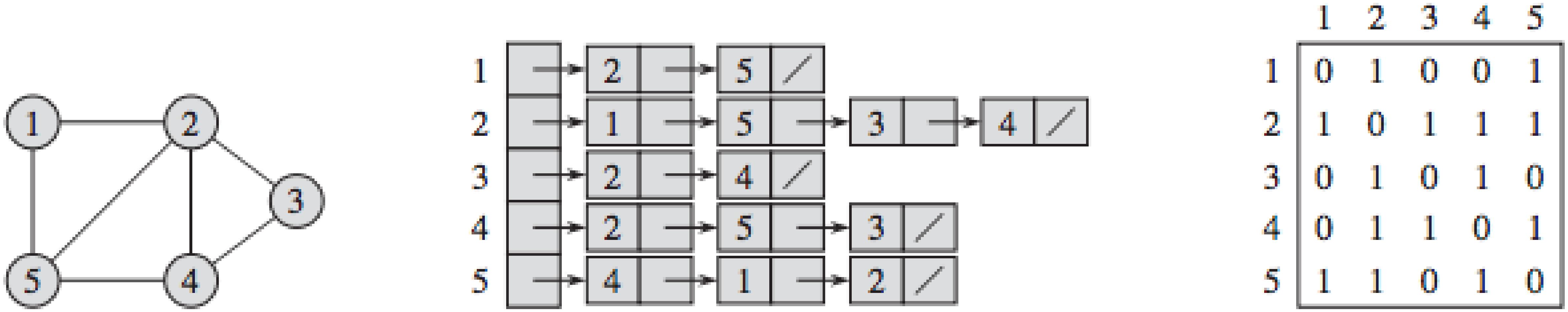


	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

Adjacency List

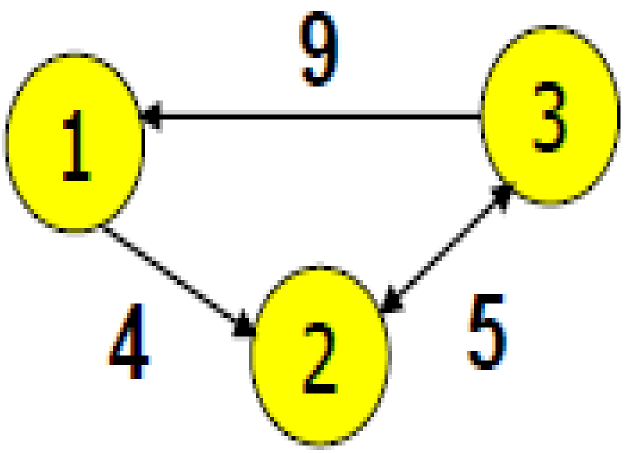
- An Adjacency list is an array of lists, each list showing the vertices a given vertex is adjacent to.
- Each vertex is represented by a linked list.
- In this representation, every vertex of a graph contains list of its adjacent vertices.

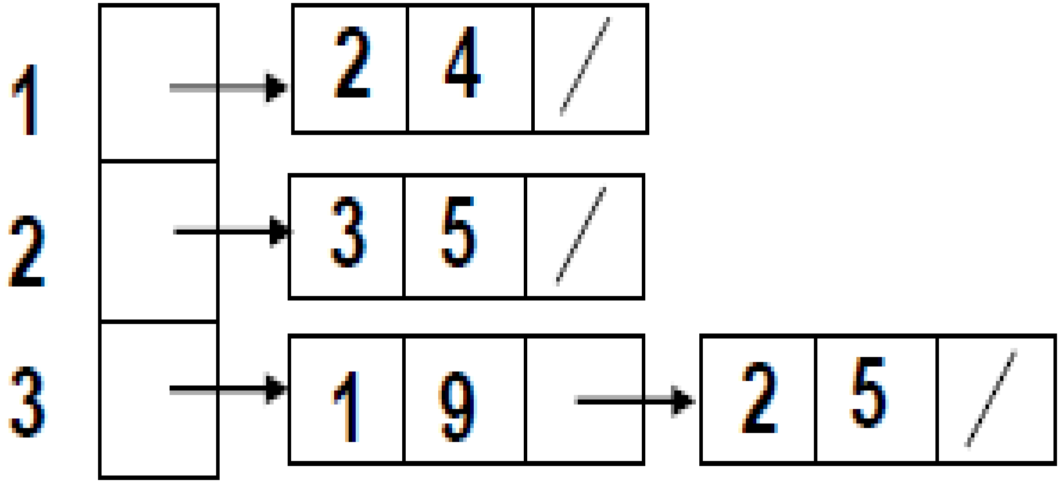
Adjacency List - Example



(a) (b) (c)

Adjacency List of a Weighted Graph





Graph Traversals

- Graph traversals provide a way to search for a vertex in a graph.
- They can be used to decide the order of vertices to be visited in the search process.
- There are two graph traversal techniques;
  1. Depth First Search
  2. Breadth First Search

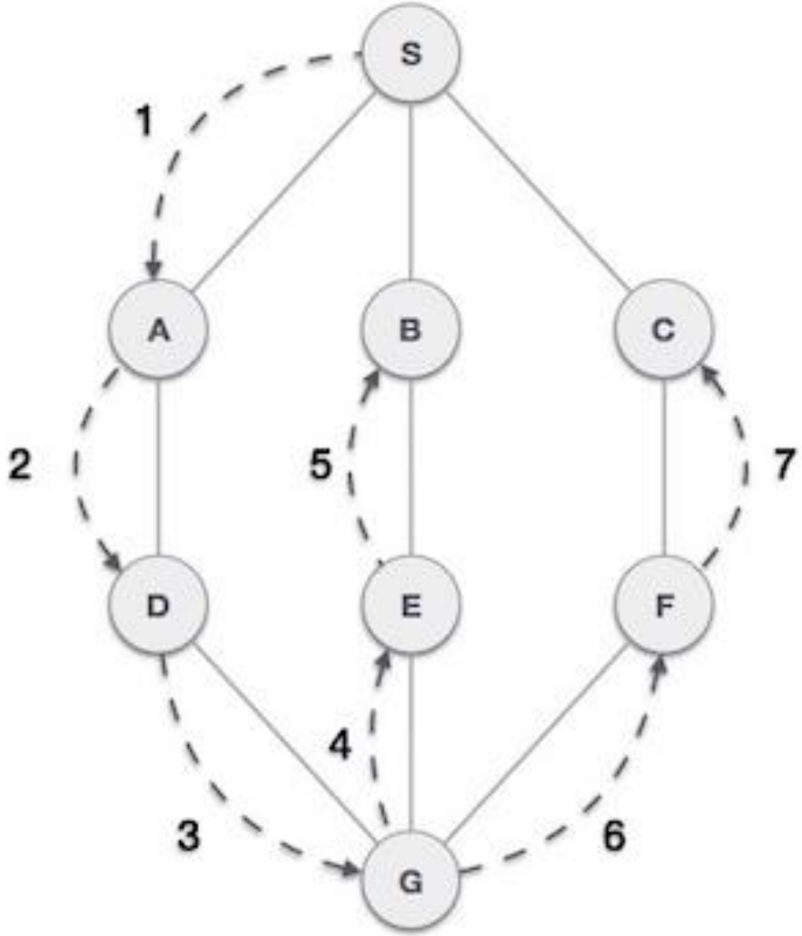
Depth-First-Search (DFS)

- Travel as far as you can down a path
- DFS can be implemented efficiently using a stack
- Produces a spanning tree as the final result after traversal
- Spanning tree: a sub-graph of an undirected connected graph, which includes all the vertices of the graph with a minimum possible number of edges

Depth-First-Search (DFS) : Steps

1. Start from an arbitrary node to start traversal
2. Visit (Explore) an unvisited adjacent edge
3. If the node visited is a dead end, go back to the previous node (Backtrack)
4. Stop when no unvisited nodes are found and no backtracking can be done

Depth-First-Search (DFS) - Example



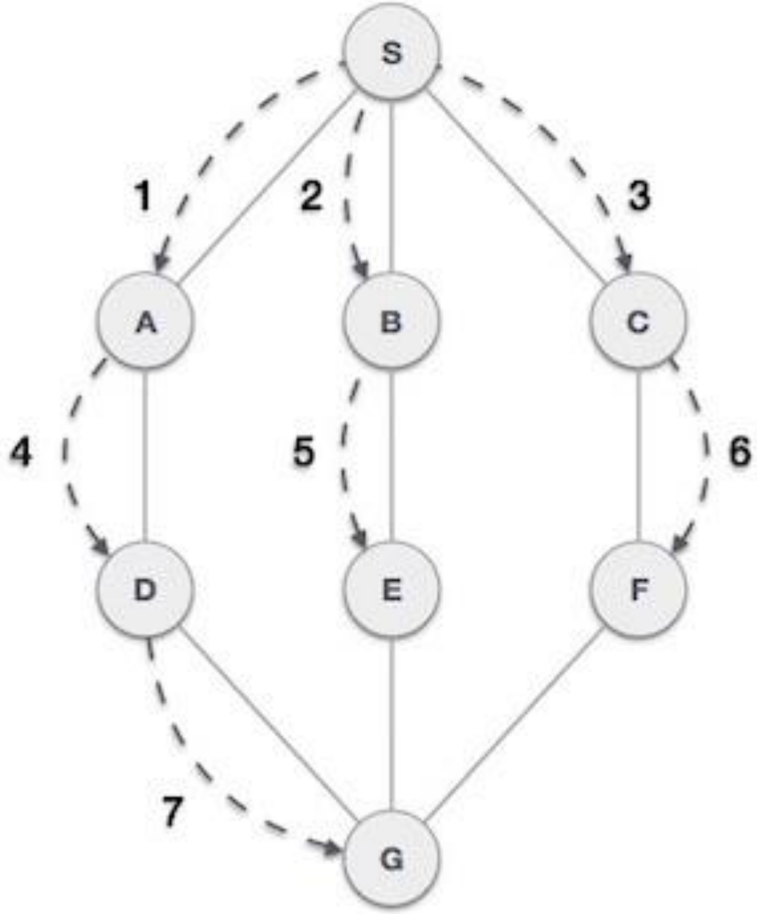
Breadth-First-Search (BFS)

- Look at all possible paths at the same depth before you go at a deeper level
- BFS can be implemented efficiently using a queue
- Produces a spanning tree as the final result after traversal

Breadth-First-Search : Steps

- Starts from a node
- Visit neighbouring nodes
- For each nearest node, explores their unexplored neighbour nodes until it finds the goal

Breadth-First-Search (BFS) - Example

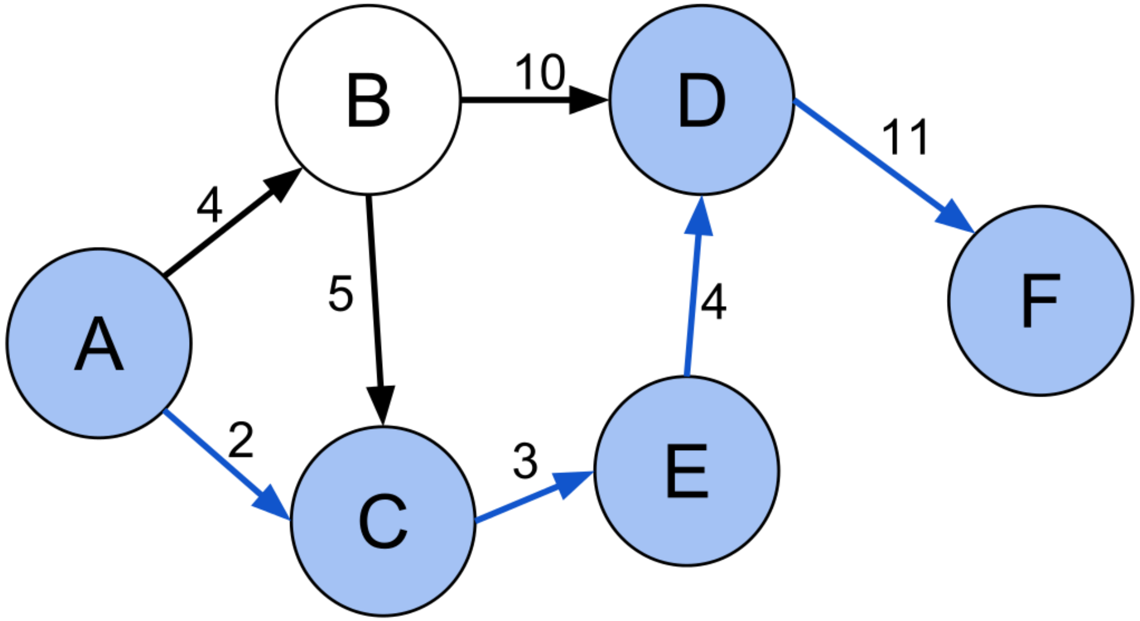


Applications of Graphs

- Undirected graphs can be used in social networks like LinkedIn, Facebook.
- Undirected weighted graphs can be used to represent transportation networks among cities.
- Directed graphs can be used to represent relationship among web pages.(World Wide Web)
- To model electrical circuits.
- To model computer network ,nodes are workstations and the edges are network connections.

Shortest Path

- The shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.
- Algorithms: Dijkstra’s Algorithm, Kruskal's Algorithm, Prim’s Algorithm



Questions?