GRAPH INTRODUCTION

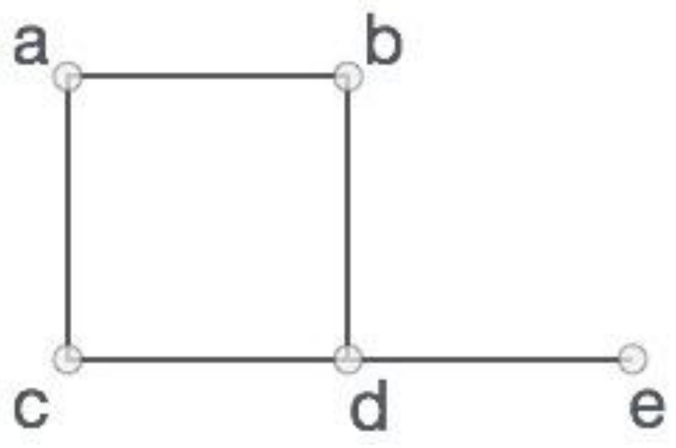
INTRODUCTION

- A graph data structure is a collection of nodes that have data and are connected to other nodes.
- A graph is a pictorial representation of a set of objects
- where some pairs of objects are connected by links.
- The interconnected objects are represented by points termed as vertices
- the links that connect the vertices are called edges

from the graph,

V = {a, b, c, d, e}

E = {ab, ac, bd, cd, de}



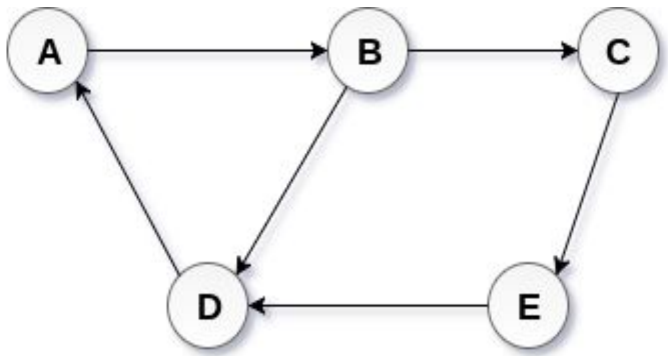
TYPES OF GRAPH

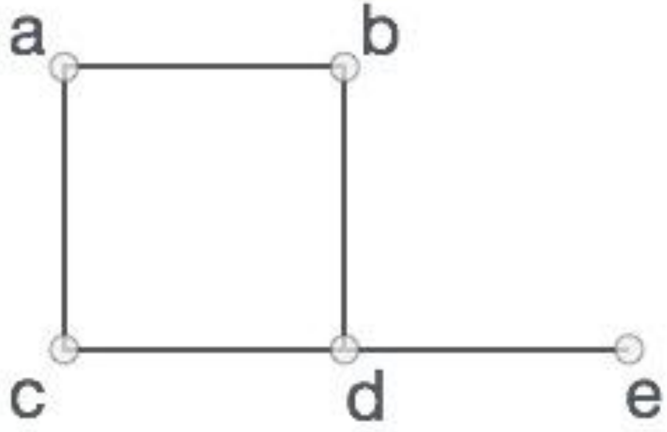
(1) Undirected Graph

- edges are not associated with the directions with them.
- If an edge exists between vertex A and B then the vertices can be traversed from B to A as well as A to B.
- Bidirectional

Directed Graph

- edges form an ordered pair.
- Edges represent a specific path from some vertex A to another vertex B.
- Node A is called initial node while node B is called terminal node.
- One way relationship not backwards
- also called as di graph





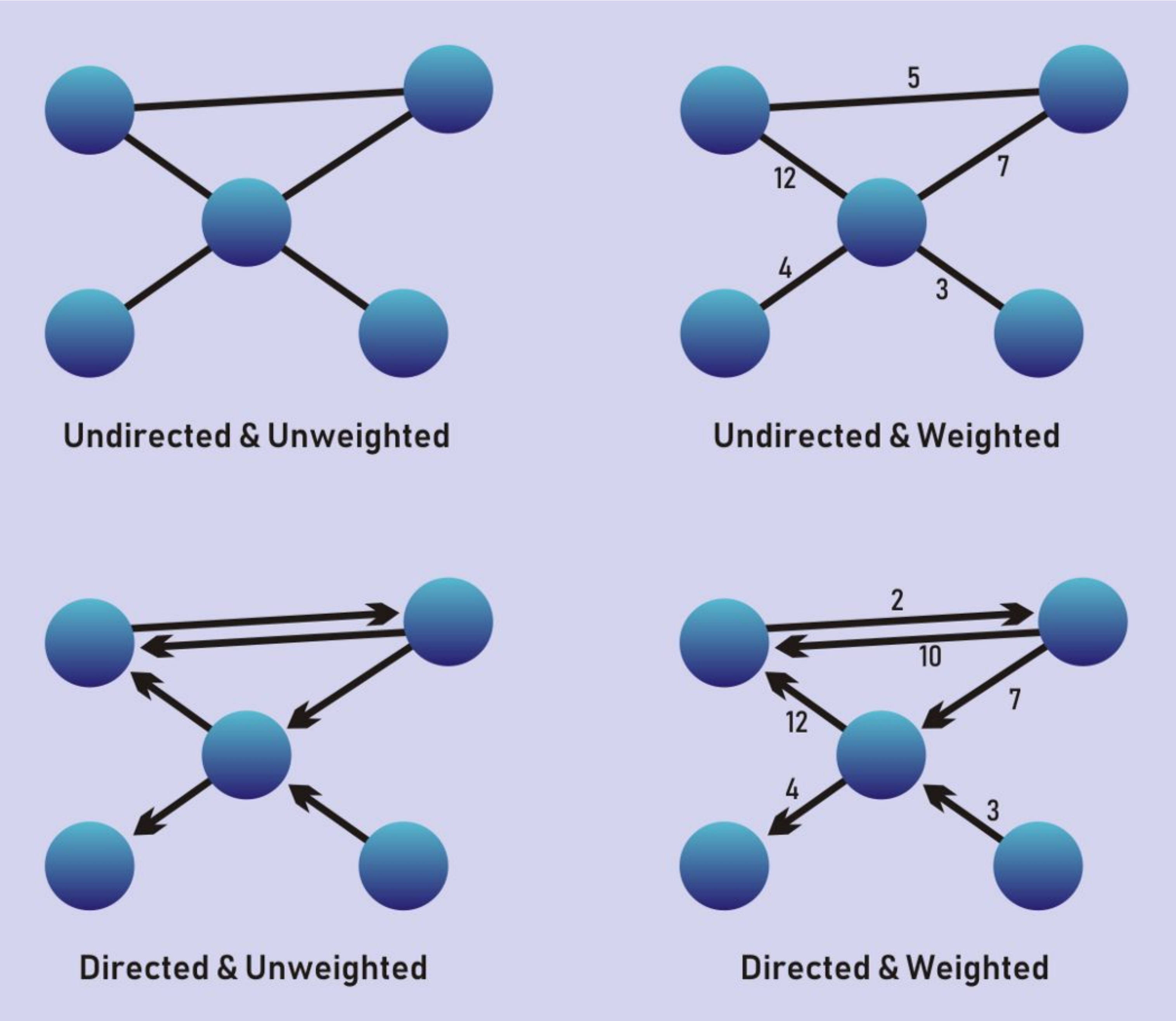
(2) WEIGHTED GRAPH

Edges have some weight

UNWEIGHTED GRAPH

Edges does not have weight

- **Undirected & Unweighted:** relationships do not have weight and are bidirectional.
- **Undirected & Weighted:** relationships have a weight and are bidirectional.
- **Directed & Unweighted:** relationships do not have weight and are one way.
- **Directed & Weighted:** relationships have a weight and are one way.



GRAPH TERMINOLOGY

Path

A path can be defined as the sequence of nodes that are followed in order to reach some terminal node from the initial node.

Closed Path

A path will be called as closed path if the initial node is same as terminal node. A path will be closed path if $V_0=V_N$.

Cycle

A cycle can be defined as the path which has no repeated edges or vertices except the first and last vertices.

Connected Graph

- A connected graph is the one in which some path exists between every two vertices (u, v) in V.
- There are no isolated nodes in connected graph.

Complete Graph

- A complete graph is the one in which every node is connected with all other nodes.
- A complete graph contain n(n-1)/2 edges where n is the number of nodes in the graph.

Weighted Graph

- In a weighted graph, each edge is assigned with some data such as length or weight.
- The weight of an edge e can be given as w(e) which must be a positive (+) value indicating the cost of traversing the edge.

Digraph

- A digraph is a directed graph in which each edge of the graph is associated with some direction
- traversing can be done only in the specified direction.

Loop

An edge that is associated with the similar end points can be called as Loop.

Adjacent Nodes

- If two nodes u and v are connected via an edge e, then the nodes u and v are called as neighbors or adjacent nodes.

Degree of the Node

- A degree of a node is the number of edges that are connected with that node.
- A node with degree 0 is called as isolated node.

BASIC OPERATIONS

- Add Vertex
- Add Edge
- Display Vertex
- Graph traversal

GRAPH REPRESENTATION

1. ADJACENCY MATRIX

- An adjacency matrix is a array of V x V vertices.
- Each row and column represent a vertex.
- If the value of any element a[i][j] is 1, it represents that there is an edge connecting vertex i and vertex j.
- Edge lookup ie.,checking if an edge exists between vertex A and vertex B is extremely fast but it requires more space.
- Dense graph

2. ADJACENCY LIST

- It represents a graph as an array of linked lists.
- The index of the array represents a vertex and each element in its linked list represents the other vertices that form an edge with the vertex.
- Sparse Graph
- An adjacency list is efficient in terms of storage because only need to store the values for the edges.
- For a graph with millions of vertices, this can mean a lot of saved space.

SPACE COMPLEXITY:

Adjacent matrix= O(n^2)

Adacency list = O(n+2e)

APPLICATIONS OF GRAPH

- **Computer science** the flow of computation.
- **Google maps**building transportation systems,
  - where intersection of two(or more) roads are considered to be a vertex and the road connecting two vertices is considered to be an edge,
  - their navigation system is based on the algorithm to calculate the shortest path between two vertices.
- In **Facebook** Friend suggestion algorithm
  - users are considered to be the vertices and if they are friends then there is an edge running between them.
  - Facebook is an example of **undirected graph**.
- In **World Wide Web** Google page ranking system
  - web pages are considered to be the vertices.
  - There is an edge from a page u to other page v if there is a link of page v on page u. This is an example of **Directed graph**.
- In **Operating System** Resource Allocation Graph
  - each process and resources are considered to be vertices.
  - Edges are drawn from resources to the allocated process, or from requesting process to the requested resource.
  - If this leads to any formation of a cycle then a deadlock will occur.