### Graph:

## **Basic Terminologies and Representations of Graph:**

A Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a <u>Graph</u> is composed of a set of vertices(V) and a set of edges(E). The graph is denoted by G(E, V).

# **Components of a Graph**

- Vertices: Vertices are the fundamental units of the graph. Sometimes, vertices are also known as vertex or nodes. Every node/vertex can be labeled or unlabelled.
- Edges: Edges are drawn or used to connect two nodes of the graph. It can be ordered pair of nodes in a directed graph. Edges can connect any two nodes in any possible way. There are no rules. Sometimes, edges are also known as arcs. Every edge can be labeled/unlabelled.



# Types Of Graph

# 1. Null Graph

A graph is known as a null graph if there are no edges in the graph.

# 2. Trivial Graph

Graph having only a single vertex, it is also the smallest graph possible.



# 3. Undirected Graph

A graph in which edges do not have any direction. That is the nodes are unordered pairs in the definition of every edge.

# 4. Directed Graph

A graph in which edge has direction. That is the nodes are ordered pairs in the definition of every edge.



### 5. Connected Graph

The graph in which from one node we can visit any other node in the graph is known as a connected graph.

### 6. Disconnected Graph

The graph in which at least one node is not reachable from a node is known as a disconnected graph.



### 7. Regular Graph

The graph in which the degree of every vertex is equal to the other vertices of the graph. Let the degree of each vertex be  $\mathbf{K}$  then the graph

## 8. Complete Graph

The graph in which from each node there is an edge to each other node.



### 9. Cycle Graph

The graph in which the graph is a cycle in itself, the degree of each vertex is 2.

# 10. Cyclic Graph

A graph containing at least one cycle is known as a Cyclic graph.



# 11. Directed Acyclic Graph

A Directed Graph that does not contain any cycle.

### 12. Bipartite Graph

A graph in which vertex can be divided into two sets such that vertex in each set does not contain any edge between them.



# 13. Weighted Graph

- A graph in which the edges are already specified with suitable weight is known as a weighted graph.
- Weighted graphs can be further classified as directed weighted graphs and undirected weighted graphs.

# Tree v/s Graph

Trees are the restricted types of graphs, just with some more rules. Every tree will always be a graph but not all graphs will be trees. <u>Linked List</u>, <u>Trees</u>, and <u>Heaps</u> all are special cases of graphs.



### **Representation of Graphs**

There are two ways to store a graph:

- Adjacency Matrix
- Adjacency List

676

#### <u>Adjacency Matrix</u>

In this method, the graph is stored in the form of the 2D matrix where rows and columns denote vertices. Each entry in the matrix represents the weight of the edge between those vertices.



### <u>Adjacency List</u>

This graph is represented as a collection of linked lists. There is an array of pointer which points to the edges connected to that vertex.



<u>Comparison between Adjacency Matrix and Adjacency List</u> When the graph contains a large number of edges then it is good to store it as a matrix because only some entries in the matrix will be empty. An algorithm such as <u>Prim's</u> and <u>Dijkstra</u> adjacency matrix is used to have less complexity.

Action	Adjacency Matrix	Adjacency List
Adding Edge	O(1)	O(1)
Removing and edge	O(1)	O(N)

	679	
Initializing	O(N*N)	O(N)

#### **Basic Operations on Graphs**

Below are the basic operations on the graph:

- Insertion of Nodes/Edges in the graph Insert a node into the graph.
- Deletion of Nodes/Edges in the graph Delete a node from the graph.
- Searching on Graphs Search an entity in the graph.
- Traversal of Graphs Traversing all the nodes in the graph.

### **Usage of graphs**

- Maps can be represented using graphs and then can be used by computers to provide various services like the shortest path between two cities.
- When various tasks depend on each other then this situation can be represented using a Directed Acyclic graph and we can find the order in which tasks can be performed using topological sort.
- State Transition Diagram represents what can be the legal moves from current states. In-game of tic tac toe this can be used.

### **Real-Life Applications of Graph**



#### Graph search:

A Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph is composed of a set of vertices( V) and a set of edges(E). The graph is denoted by G(E, V).



### **Components of a Graph**

- Vertices: Vertices are the fundamental units of the graph. Sometimes, vertices are also known as vertex or nodes. Every node/vertex can be labeled or unlabelled.
- Edges: Edges are drawn or used to connect two nodes of the graph. It can be ordered pair of nodes in a directed graph. Edges can connect any two nodes in any possible way. There are no rules. Sometimes, edges are also known as arcs. Every edge can be labeled/unlabelled.

Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like linkedIn, Facebook. For example,

681