



# Data Structures [R1UC308B]

Module-IX: Graph

**Dr. Subhash Chandra Gupta**



School of Computer Science and Engineering  
Plat No 2, Sector 17A, Yamuna Expressway  
Greater Noida, Uttar Pradesh - 203201

July 31, 2025



# Contents

Graph	3
Introduction	3
Types of Graphs	5
Representation	20
Applications	23



## Introduction

- ▶ 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.
- ▶ The graph is denoted by  $G(V, E)$ .
- ▶ Where  $V$  is a set of vertices and  $E$  is a set of edges.

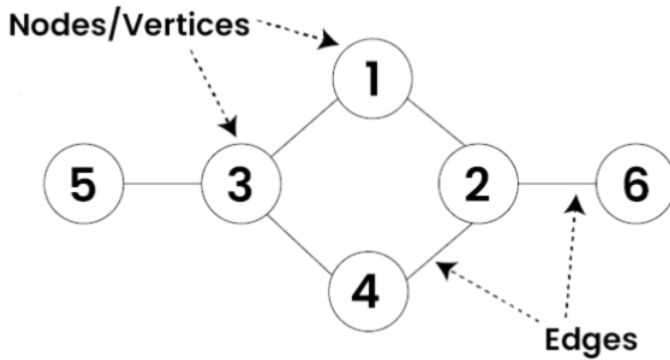


Figure: Graph



## Types of Graphs

- **Null Graph:** A graph is known as a null graph if there are no edges in the graph.

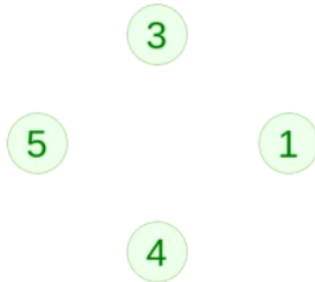


Figure: Null Graph



- ▶ **Trivial Graph:** Graph having only a single vertex, it is also the smallest graph possible.



Figure: Trivial Graph



- **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.

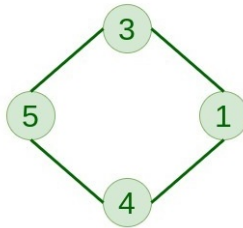


Figure: Undirected Graph



- **Directed Graph:** A graph in which edge has direction. That is the nodes are ordered pairs in the definition of every edge.

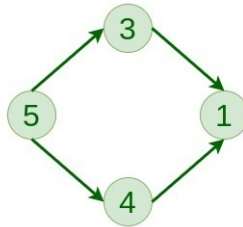


Figure: Directed Graph





- **Connected Graph:** The graph in which from one node we can visit any other node in the graph is known as a connected graph.

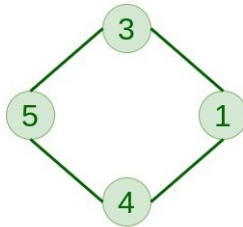


Figure: Connected Graph



- **Disconnected Graph:** The graph in which at least one node is not reachable from a node is known as a disconnected graph.

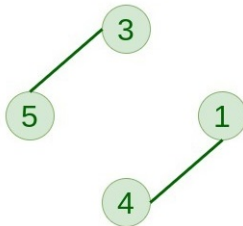


Figure: Disconnected Graph



- **Regular Graph:** The graph in which the degree of every vertex is equal to  $K$  is called  $K$  regular graph. The degree of a vertex in a graph is the number of edges that are connected to it. It can also be defined as the number of neighbors or vertices that are adjacent to a vertex.

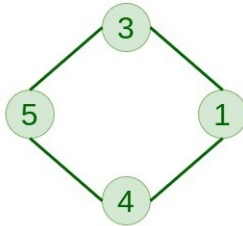


Figure: Regular Graph



- **Complete Graph:** The graph in which from each node there is an edge to each other node.

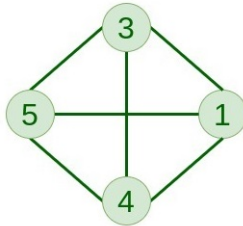


Figure: Complete Graph



- **Cycle Graph:** The graph in which the graph is a cycle in itself, the minimum value of degree of each vertex is 2.

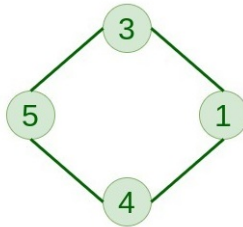


Figure: Cycle Graph



- **Cyclic Graph:** A graph containing at least one cycle is known as a Cyclic graph.

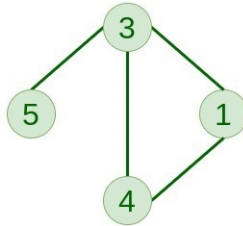


Figure: Cyclic Graph



- **Directed Acyclic Graph:** A Directed Graph that does not contain any cycle.

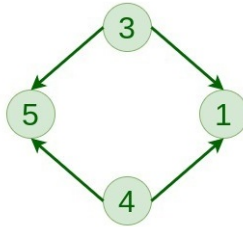


Figure: Directed Acyclic Graph



- **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.

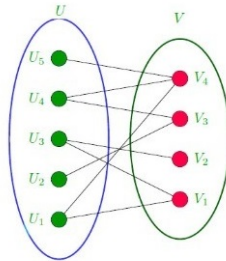
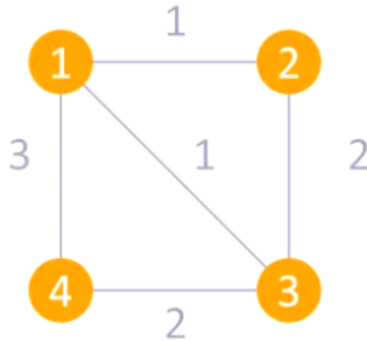


Figure: Bipartite Graph





- **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**.



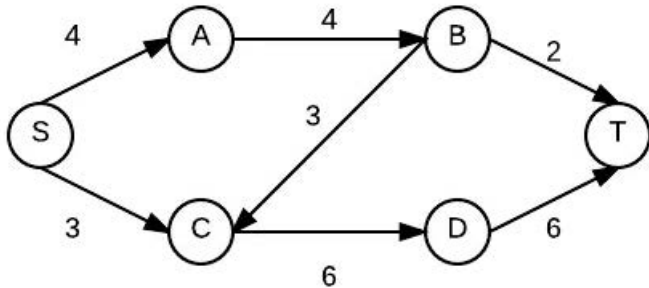


Figure: Weighted Graph



## Representation

- Adjacency Matrix using 2D-array

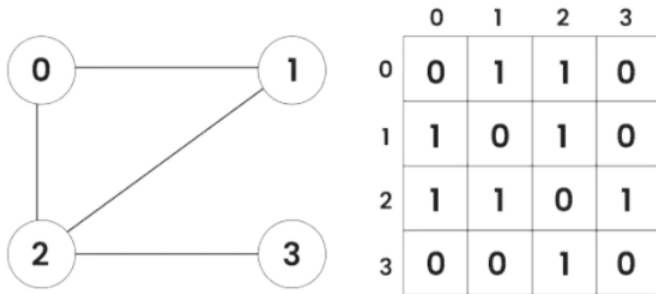


Figure: Adjacency Matrix

- Rows and columns of the matrix represent vertices.



- ▶ Each entry in the matrix represents the weight of the edge between those vertices.
- ▶  $M[i][j]=k$  represents edge between nodes  $i$  and  $j$  having weight  $k$  in the weighted graph and  $k=1$  if graph is unweighed.
- ▶ For example  $M[2][3]=1$  shows there is a edge between node 2 and 3.
- ▶ 0 means there is no direct edge.



► Adjacency List using 1D-array of Nodes and Linked List

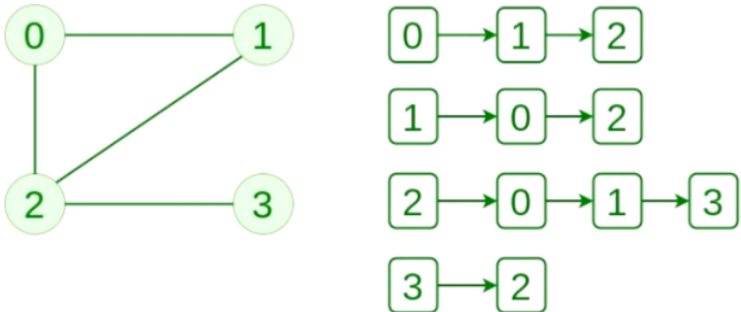


Figure: Adjacency List



# Applications

## Applications

- ▶ Social media analysis
- ▶ Network monitoring
- ▶ Financial trading
- ▶ Internet of Things (IoT) management
- ▶ Autonomous vehicles
- ▶ Disease surveillance
- ▶ Route finding
- ▶ **Example:** The best example of graphs in the real world is Facebook. Each person on Facebook is a node and is connected through edges. Thus, A is a friend of B. B is a friend of C, and so on.



## Advantages of Graph:

- ▶ Representing complex data
- ▶ Efficient data processing
- ▶ Network analysis
- ▶ Path-finding
- ▶ Visualization
- ▶ Machine learning

## Disadvantages of Graph:

- ▶ Limited representation
- ▶ Difficulty in interpretation
- ▶ Scalability issues
- ▶ Data quality issues
- ▶ Lack of standardization
- ▶ Privacy concerns





# Thank you

Please send your feedback or any queries to  
[subhash.chandra@galgotiasuniversity.edu.in](mailto:subhash.chandra@galgotiasuniversity.edu.in)