Modern Education Society's Wadia College of Engineering, Pune

210256: DATA STRUCTURES and ALGORITHM LABORATORY (2019 C0URSE)

NAME OF STUDENT:	**CLASS:**
SEMESTER/YEAR:	**ROLL NO:**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED BY:	**EXPERIMENT NO: C13**

TITLE: REPRESENTATION OF ADJACENCY LIST AND MATRIX OF A GRAPH

AIM/PROBLEM STATEMENT: Represent a given graph using adjacency matrix/ list to perform DFS and using list to perform BFS. Use the map of the area around the college as the graph. Identify the prominent land marks as nodes and perform DFS and BFS on that

OBJECTIVES:

1. To understand graph data structure using adjacency matrix/ list
2. Able to perform graph traversal DFS using adjacency matrix with the help of stack 3. 3. Able to perform graph traversal BFS using adjacency list with the help of Queue

OUTCOMES:

1. Apply and analyze linear data structures to solve non-linear data structure problems.

PRE-REQUISITE:

1. Basic Knowledge of 2D array
2. Basic Knowledge of Linked list, Stack and Queue

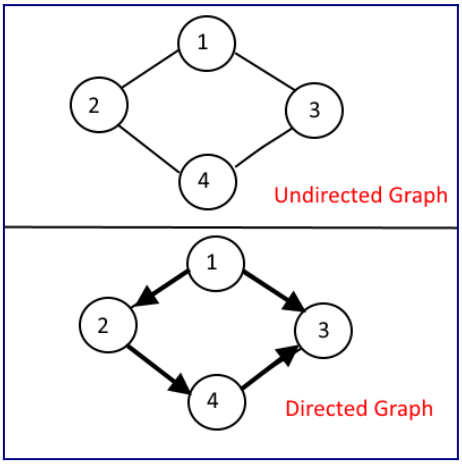
THEORY:

$$G = (V,E)$$

Graph is a collection of nodes or vertices (V) and edges(E) between them. We can traverse these nodes using the edges. These edges might be weighted or non-weighted.

There can be two kinds of Graphs

- Un-directed Graph – when you can traverse either direction between two nodes.
- Directed Graph – when you can traverse only in the specified direction between two nodes.

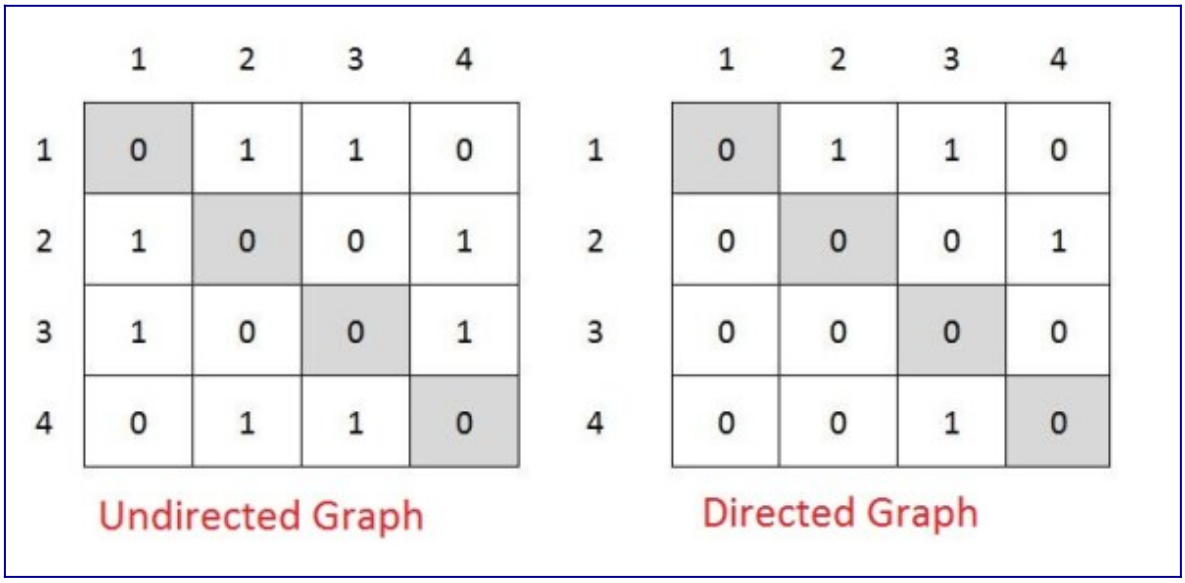


There are two common ways to represent a Graph:

- Adjacency Matrix
- Adjacency List

Adjacency Matrix:

Adjacency Matrix is 2-Dimensional Array which has the size VxV, where V are the number of vertices in the graph. See the example below, the Adjacency matrix for the graph shown above.



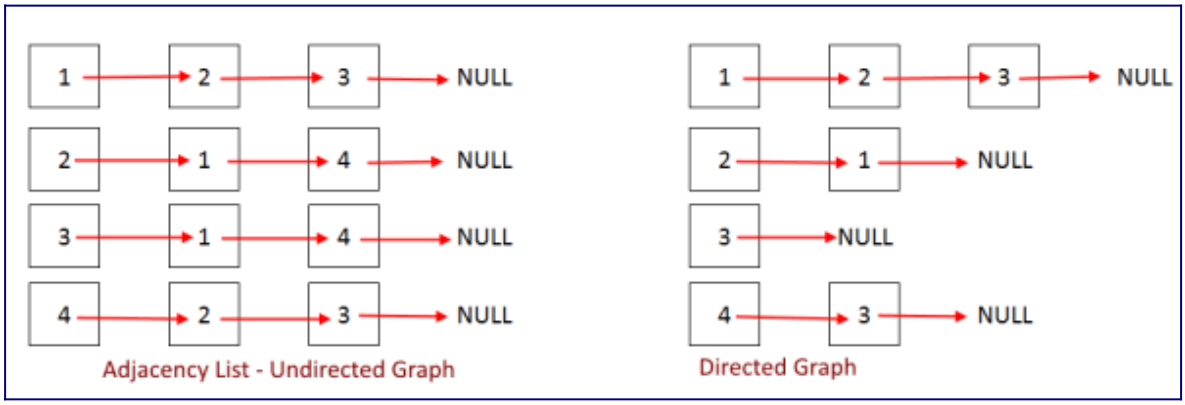
adjMaxtrix[i][j] = 1 when there is edge between Vertex i and Vertex j, else 0.

It's easy to implement because removing and adding an edge takes only O(1) time.

But the drawback is that it takes $O(V^2)$ space even though there are very less edges in the graph.

Adjacency List:

Adjacency List is the Array[] of Linked List, where array size is same as number of Vertices in the graph. Every Vertex has a Linked List. Each Node in this Linked list represents the reference to the other vertices which share an edge with the current vertex. The weights can also be stored in the Linked List Node.



The code may look complex since everything is being implemented from the scratch like linked list and so on. For better understanding, read more articles for easier implementations (Adjacency Matrix and Adjacency List)

QUESTIONS:

1) List the applications of graph?

2) Given an undirected graph G with V vertices and E edges, What is the sum of the degrees of all vertices?