

St. Francis Institute of Technology, Mumbai-400 103
Department of Information Technology

A.Y. 2023-24

Class: SE-ITA/B, Semester: III

Subject: DATA STRUCTURE LAB

Experiment – 5 Graph using adjacency matrix

1. Aim: Write a C program to implement a graph using adjacency matrix representation.

2. Objectives: After study of this experiment, the student will be able to

- To use basic principles of programming as applied to complex data structures
- To learn fundamentals of graphs

3. Outcomes: After study of this experiment, the student will be able to

- Implement a graph using adjacency matrix and understand its operations
- Understand the concepts and apply the methods in graphs.

4. Prerequisite: Graphs, Types of Graphs.

5. Requirements: PC and Turbo C compiler version 3.0 , Codeblock20.03

6. Pre-Experiment Exercise:

Brief Theory:

A. Graphs

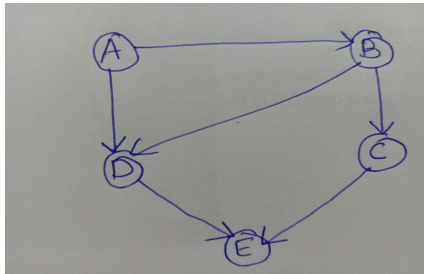
- A graph is basically, a collection of vertices (also called nodes) and edges that connect these vertices.
- A graph is often viewed as a generalization of the tree structure, where instead of a having a purely parent-to-child relationship between tree nodes, any kind of complex relationships between the nodes can be represented.

Why graphs are useful?

- Graphs are widely used to model any situation where entities or things are related to each other in pairs; for example, the following information can be represented by graphs:
- Family trees in which the member nodes have an edge from parent to each of their children.
- Transportation networks in which nodes are airports, intersections, ports, etc. The edges can be airline flights, one-way roads, shipping routes, etc. Definition
- A graph G is defined as an ordered set (V, E) , where $V(G)$ represent the set of vertices and $E(G)$ represents the edges that connect the vertices.

- The figure given shows a graph with $V(G) = \{ A, B, C, D \text{ and } E \}$ and

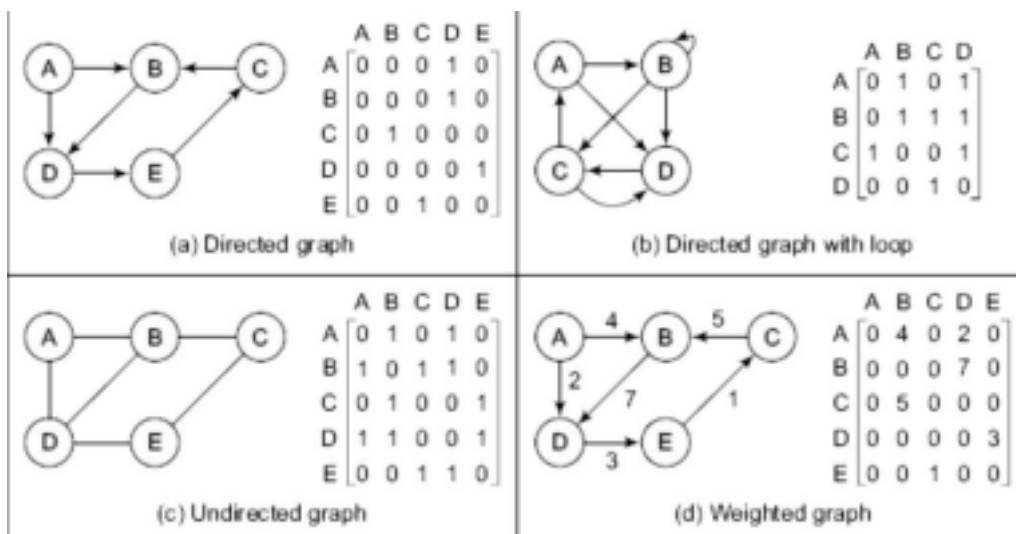
$$E(G) = \{ (A, B), (B, C), (A, D), (B, D), (D, E), (C, E) \}.$$



B. Representation of Graphs

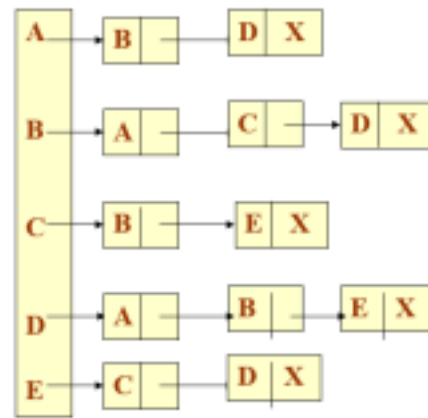
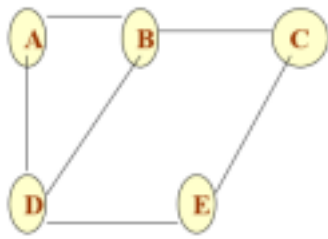
1. Adjacency Matrix representation

- An adjacency matrix is used to represent which nodes are adjacent to one another. By definition, we have learnt that, two nodes are said to be adjacent if there is an edge connecting them.
- In a directed graph G , if node v is adjacent to node u , then surely there is an edge from u to v . That is, if v is adjacent to u , we can get from u to v by traversing one edge. For any graph G having n nodes, the adjacency matrix will have dimensions of $n \times n$.



2. Adjacency List representation

- The adjacency list is another way in which graphs can be represented in computer's memory.
- This structure consists of a list of all nodes in G .
- Furthermore, every node is in turn linked to its own list that contains the names of all other nodes that are adjacent to itself.



7. Laboratory Exercise

A. Procedure

1. Write a C program to implement a Directed/Undirected Graph using adjacency matrix and show all the following operations in switch case,
 - i) Create a graph
 - ii) Display graph

B. Result/Observation/Program code:

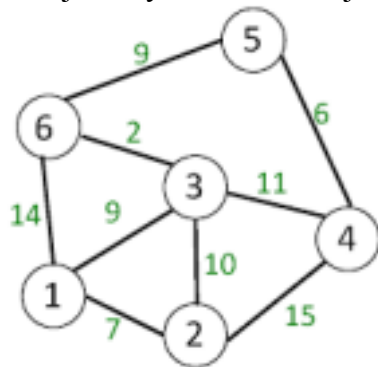
Observe the output for the above code and print it.

8. Post-Experiments Exercise

A. Questions:

A. Explain Directed and Weighted Graph

B. Show the adjacency matrix and adjacency list representation for the graph given below.



B. Conclusion:

1. Summary of Experiment
2. Importance of Experiment

C. References:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Reema Thareja; Data Structures using C; Oxford.
3. Data Structures A Pseudocode Approach with C, Richard F. Gilberg & Behrouz A. Forouzan, second edition, CENGAGE Learning.