EC-200 Data Structures

Lab Manual 10

**Course Instructor:** Lecturer Anum Abdul Salam

**Lab Engineer:** Lab Engineer Ayesha Batool

**Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Degree/ Syndicate: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Trait** | **Obtained Marks** | **Maximum Marks** |
| **R1** | **Application Functionality 20%** |  | 20 |
| **R2** | **Specification & Data structure implementation**  **30%** |  | 30 |
| **R3** | **Reusability**  **10%** |  | 10 |
| **R4** | **Input Validation**  **10%** |  | 10 |
| **R5** | **Efficiency**  **20%** |  | 20 |
| **R6** | **Delivery**  **10%** |  | 10 |
| **R7** | **Plagiarism above 80%** |  | 1 |
|  | **Total** |  | 10 |

**Total Marks = O**𝒃𝒕𝒂𝒊𝒏𝒆𝒅 𝑴𝒂𝒓𝒌𝒔 (∑6𝟏 𝑹𝒊 ∗ 𝑹7)

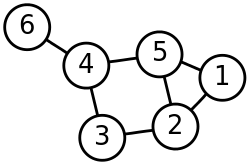
# LAB # 10: Graph Implementation

## Lab Objective:

To Implement graph using adjacency list

## Lab Description:

In mathematics graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices, nodes, or points which are connected by edges, arcs, or lines. A graph data structure consists of a finite (and possibly mutable) set of vertices or nodes or points, together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph.



**Figure 10.1:** Data in graph

****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, and the links that connect the vertices are called edges. Formally, a graph is a pair of sets (V, E), where V is the set of vertices and E is the set of edges, connecting the pairs of vertices. Take a look at the following graph.

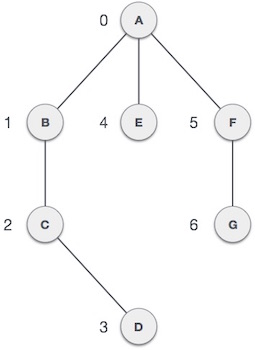
**Figure 10.2:** Edge and vertex based representation of graph

In the above graph, V = {a, b, c, d, e}, E = {ab, ac, bd, cd, de}

**Graph terminology**

Mathematical graphs can be represented in data structure. We can represent a graph using an array of vertices and a two-dimensional array of edges. Before we proceed further, let's familiarize ourselves with some important terms.

* **Vertex** − Each node of the graph is represented as a vertex. In the following example, the labeled circle represents vertices. Thus, A to G are vertices. We can represent them using an array as shown in the following image. Here A can be identified by index 0. B can be identified using index 1 and so on.
* **Edge** − Edge represents a path between two vertices or a line between two vertices. In the following example, the lines from A to B, B to C, and so on represents edges. We can use a two-dimensional array to represent an array as shown in the following image. Here AB can be represented as 1 at row 0, column 1, BC as 1 at row 1, column 2 and so on, keeping other combinations as 0.
* **Adjacency** − Two node or vertices are adjacent if they are connected to each other through an edge. In the following example, B is adjacent to A, C is adjacent to B, and so on.
* **Path** − Path represents a sequence of edges between the two vertices. In the following example, ABCD represents a path from A to D.



**Figure 10.3:** Weighted Graph

1. **Graph ADT operations**

Following are the graph operations associated with graph ADT

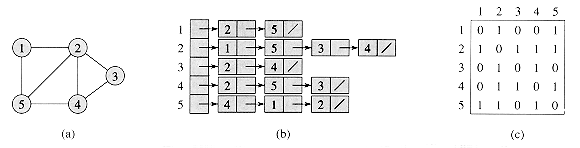
* + - **Create an empty graph**
    - **Destroy a graph**
    - **isEmpty**
    - **Determine number of vertices in a graph**
    - **Determine number of edges in a graph**
    - Determine whether an edge exists between two vertices
    - **Insert a vertex in a graph**
    - **Insert an edge between two vertices**
    - **Delete a vertex from graph**
    - **Delete an edge between two vertices**
    - **Search a vertex from graph**
    - Breadth First Search
    - **Depth First Search**

1. **Graph Implementation**
   1. **Adjacency Matrix**

Adjacency matrix of a graph with n vertices 0,1,…,n-1 is an nxn array matrix such that matrix[i][j] is 1 (true) if there is an edge from i to j, and is 0 (false) otherwise.

* 1. **Adjacency List**

Adjacency list of a graph with n vertices 0,1,…,n-1 consists of n linked lists. The ith linked list contains a node for vertex j if and only if the graph contains an edge from i to j



**Figure 10.4:** Adjacency list and Adjacency Matrix representation of Graph

## LAB TASKS

1. Implement Adjacency list representation of Graph. Provide all the necessary functions stated above to complete the ADT.

**THINK?**

1. State Applications of Graph
2. What will be the time complexity of insertion in Adjacency list? How? Explain mathematically
3. Fill the following table. Write worst case complexity of each implementation of graph and state which the best implementation is.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Insert vertex** | **Insert edge** | **Remove vertex** | **Remove edge** | **Display graph** | **Search vertex** | **Display** | **Edge exists between vertex** |
| Adjacency List |  |  |  |  |  |  |  |  |
| Adjacency Matrix |  |  |  |  |  |  |  |  |