**Lab 12**

**Graphs**

**Introduction:**

Graph is a data structure that consists of following two components:

A finite set of vertices also called as nodes.

A finite set of ordered pair of the form (u, v) called as edge. The pair is ordered because (u, v) is not same as (v, u) in case of directed graph (di-graph). The pair of form (u, v) indicates that there is an edge from vertex u to vertex v. The edges may contain weight.

Graphs are used to represent many real life applications as they can be used to represent networks. The networks may include paths in a city or telephone network or circuit network.



**Graph representations:**

Following two are the most commonly used representations of graph.

Adjacency Matrix

Adjacency List

There are other representations also like, Incidence Matrix and Incidence List. The choice of the graph representation is situation specific. It totally depends on the type of operations to be performed and ease of use.

**Adjacency Matrix:**

Adjacency Matrix is a 2D array of size V x V where V is the number of vertices in a graph. Let the 2D array be adj[][], a slot adj[i][j] = 1 indicates that there is an edge from vertex i to vertex j. Adjacency matrix for undirected graph is always symmetric. Adjacency Matrix is also used to represent weighted graphs. If adj[i][j] = w, then there is an edge from vertex i to vertex j with weight w.



**Adjacency List:**

In graph theory and computer science, an adjacency list is a collection of unordered lists used to represent a finite graph. Each list describes the set of neighbors of a vertex in the graph. This is one of several commonly used representations of graphs for use in computer programs.



**OBJECTIVE:**

* The objective of this experiment is to implement graph using adjacency matrix and adjacency list.
* Define terms related to graphs including: vertices, edges, weighted graph, directed graph, undirected graph, adjacent vertices, path, and length of a path, cycle, simple cycle, connected vertices, connected components, adjacency set, and degree of a vertex, predecessors, and successors.
* Describe and explain standard graph symbolic representations.
* Demonstrate how to represent graphs in code using an adjacency matrix and an adjacency set.
* Discuss a basic search algorithm for graphs and contrast this search when the list of nodes to be searched is stored as a stack or a queue.
* Define **minimal spanning tree** and discuss, with diagrams, Prim's algorithm for finding the minimal spanning tree of a graph.
* Define **shortest path** and discuss, with diagrams, Dijkstra's algorithm for finding the shortest path from node x to node y of a graph.

**APPLICATION:**

* Graph is used to represent networks of **communication**, data organization, computational devices etc.
* Graph theory is also used to study molecules in chemistry and physics.
* Graph theory is also widely used in sociology.

**ISSUE:**

An invalid allocation size error occur.

**CONCLUSION:**

A graph is a popular and extensively used data structure which has many applications in the computer science field itself apart from other fields. Graphs consist of vertices and edges connecting two or more vertices.

A graph can be directed or undirected. We can represent graphs using adjacency matrix which is a linear representation as well as using adjacency linked list.