St. Francis Institute of Technology, Mumbai-400 103

Department of Information Technology

A.Y. 2020-2021

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.

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

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.

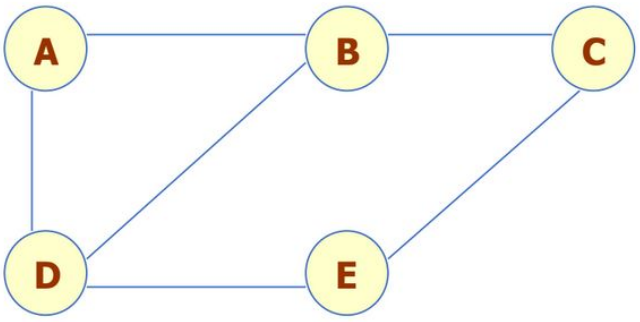
Why graphs are useful?

- 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.

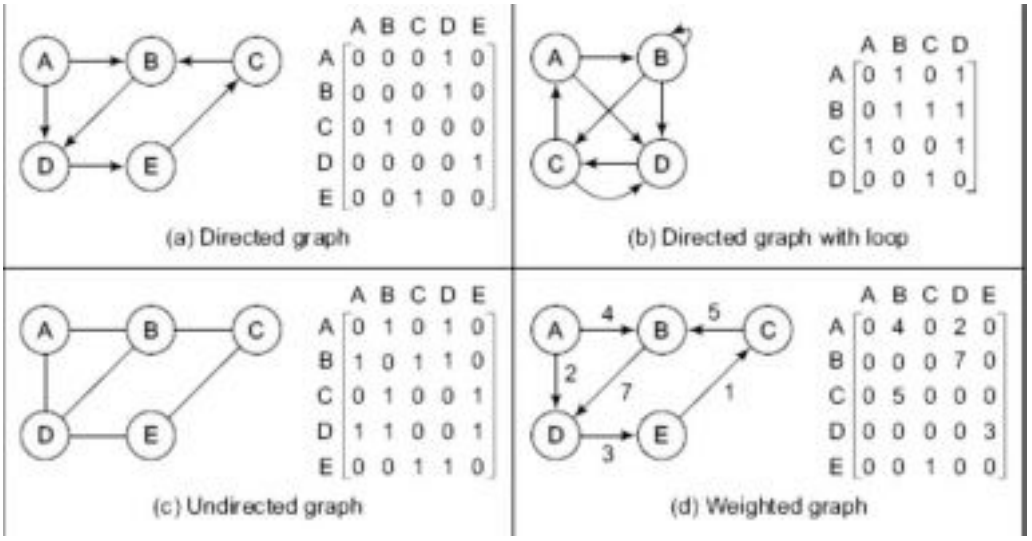
- The figure given shows a graph with
- V(G) = { A, B, C, D and E} and
- E(G) = { (A, B), (B, C), (A, D), (B, D), (D, E), (C, E) }.



B. Representation of Graphs

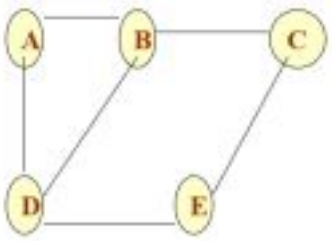
1. Adjacency Matrix representation

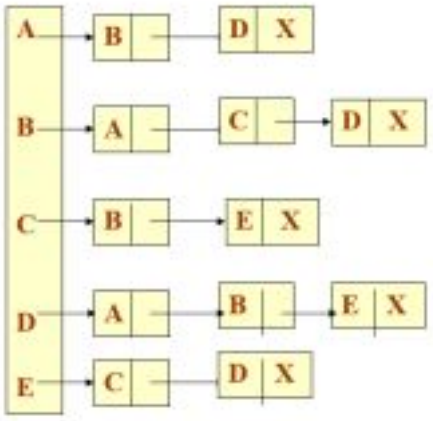
- An adjacency matrix is used to represent which nodes are adjacent to one another. By definition, we have learnt that, two nodes are said to be adjacent if there is an edge connecting them.
- In a directed graph G, if node v is adjacent to node u, then surely there is an edge from u to v. That is, if 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 X 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

Write a C program to implement a Graph using adjacency matrix and show all the following operations in switch case,

i) Create a graph

ii) Display graph

```c
// C program to implement a Graph
#include <stdio.h>
#define MAX 5
int adj[MAX][MAX];

// function declaration
void creategraph(int size);
void displaygraph(int size);


// main function
int main()
{
    int option, size;
    do
    {
        printf("\n ******MAIN MENU******* \n");
        printf("\n 1. Create a graph");
        printf("\n 2. Display graph ");
        printf("\n 3. Exit ");
        printf("\n\n Enter your option : ");
        scanf("%d", &option);
        switch(option)
        {
            case 1:
                printf("\n Enter the number of the nodes in graph : ");
                scanf("%d", &size);
```

```
                createtegraph(size);
                break;
            case 2:
                displaygraph(size);
                break;
        }
    }while(option!=3);
    return 0;
}

// function definition: createtegraph
void createtegraph(int size)
{
    int i, j;
    printf("\n Enter the adjacency matrix: ");
    for(i = 0; i < size; i++)
        for(j = 0; j < size; j++)
            scanf("%d", &adj[i][j]);

}
// function definition: createtegraph
void displaygraph(int size)
{
    int i, j;
    printf("\n Graph with adjacency matrix representation: \n");
    for(i = 0; i < size; i++)
        printf("\t%c ", i+65); // print characters in rows
    for(i = 0; i < size; i++){
        printf("\n");
        printf("%c\t",i+65); // print characters in columns
        for(j = 0; j < size; j++)
            printf("%d \t", adj[i][j]);
    }

}
```

B. Result/Observation/Program code:

Observe the output for the above code and print it.

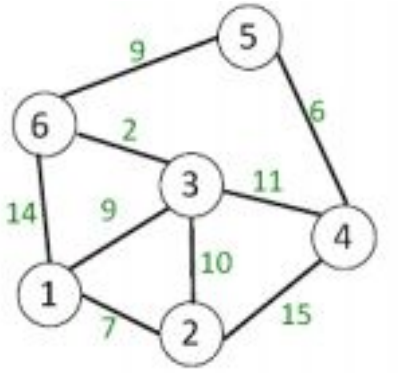
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

D:\College\DSA\Experiments\Exp5>Graph

 ******MAIN MENU*******

 1. Create a graph
 2. Display graph
 3. Exit

 Enter your option : 1

 Enter the number of the nodes in graph : 5

 Enter the adjacency matrix: 0 1 0 1 0
1 0 1 1 0
0 1 0 0 1
1 1 0 0 1
0 0 1 1 0

 ******MAIN MENU*******

 1. Create a graph
 2. Display graph
 3. Exit

 Enter your option : 2

 Graph with adjacency matrix representation:
        A       B       C       D       E
A       0       1       0       1       0
B       1       0       1       1       0
C       0       1       0       0       1
D       1       1       0       0       1
E       0       0       1       1       0
 ******MAIN MENU*******
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

******MAIN MENU*******

1. Create a graph
2. Display graph
3. Exit

Enter your option : 3

D:\College\DSA\Experiments\Exp5>
```

8. Post-Experiments Exercise

A. Questions:

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.

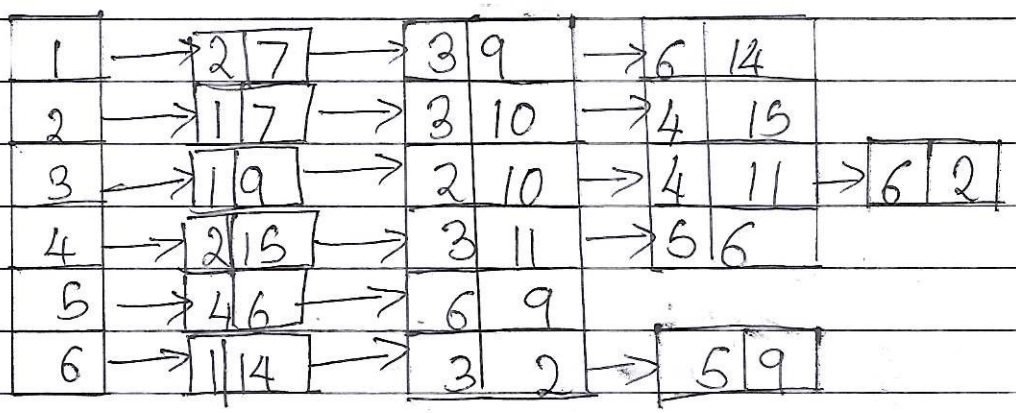
Yash Maharjan SE IT B 191061 04

Page: Date: / /

A. Questions.

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

	1	2	3	4	5	6
1	0	7	9	0	6	14
2	7	0	10	15	0	0
3	9	10	0	11	0	2
4	0	15	11	0	6	0
5	0	0	0	6	0	9
6	14	0	2	0	9	0



B. Conclusion :-

In this experiment we have implemented a graph using adjacency matrix. An adjacency matrix is used to represent which nodes are adjacent to one another. We have used a two dimentional array to represent the adjacency matrix and have written code of two functions

create graph and display ~~size~~ to create a graph and display it respectively.

Basic operations like adding an edge, removing an edge and checking weather there is an edge from vertex i to vertex j are extremely time efficient, constant time operations with adjacency matrix.