Experiment-3.2

Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities:

a. Create a Graph of N cities using Adjacency Matrix.

b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method

Student Name: Neha Sharma
UID: 20BCS4576
Branch: 20IOT1
Section/Group: A
Semester: 3rd
Date of Performance:23.11.21
Subject Name: DATA STRUCTURES LAB
Subject Code: 21O-20CSP-236_20BIT-1_A

1. Aim/Overview of the practical:

Program to Sort Create a Graph of N cities and print all nodes reachable from a given starting node in a digraph using DFS BFS method.

2. Task to be done:

To write a program for print all the nodes reachable from a given starting node in a digraph using DFS/BFS method.

3. Algorithm/Flowchart:

BFS is a traversing algorithm where we start traversing from a selected source node layerwise by exploring the neighboring nodes.
The data structure used in BFS is a queue and a graph. The algorithm makes sure that every node is visited **not more than once**.

BFS follows the following 4 steps:

1. Begin the search algorithm, by knowing the key which is to be searched. Once the key/element to be searched is decided the searching begins with the root (source) first.
2. Visit the contiguous unvisited vertex. Mark it as visited. Display it (if needed). If this is the required key, stop. Else, add it in a queue.
3. On the off chance that no neighboring vertex is discovered, expel the first vertex from the Queue.
4. Repeat step 2 and 3 until the queue is empty.

DFS follows the following 4 steps:

1. Create a recursive function that takes the index of the node and a visited array.
2. Mark the current node as visited and print the node.
3. Traverse all the adjacent and unmarked nodes and call the recursive function with the index of the adjacent node.

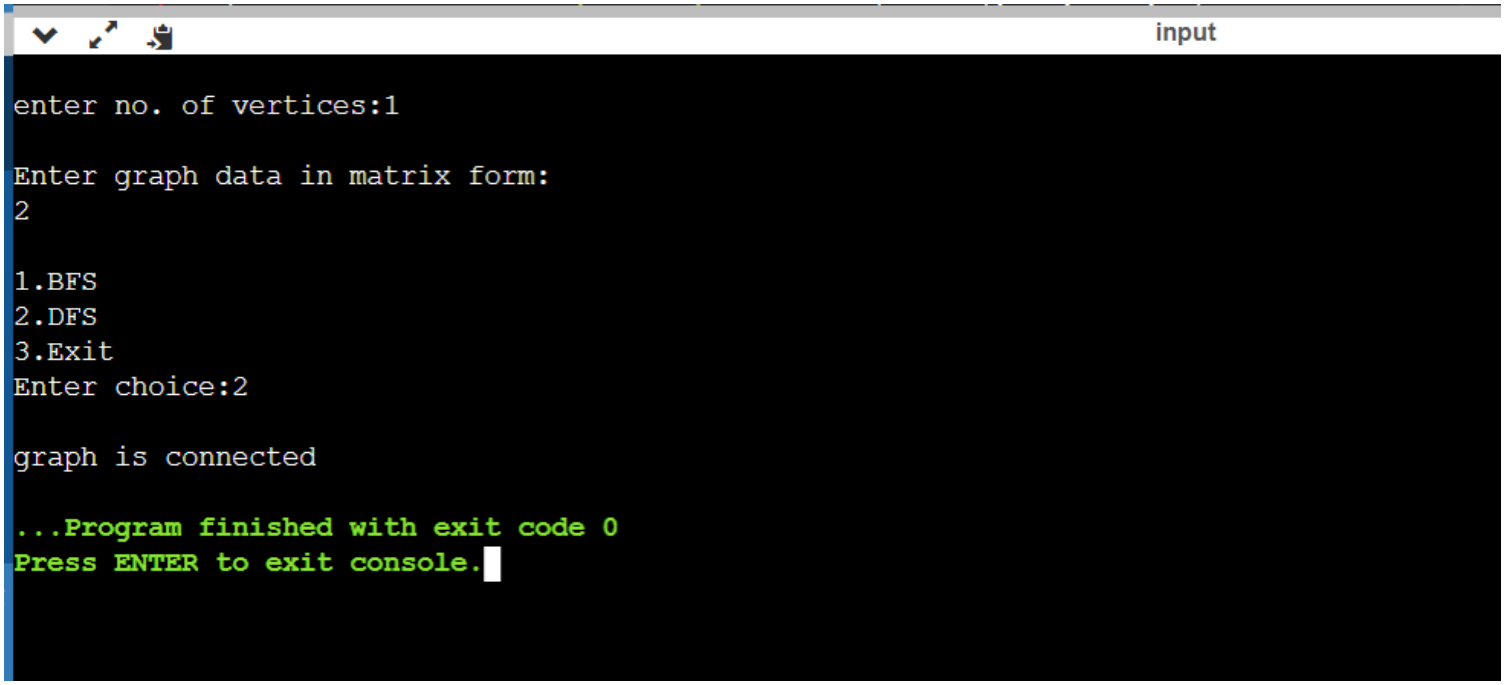
4. Code for experiment/practical:

```
#include <stdio.h>
#include <stdlib.h>
int a[20][20],q[20],visited[20],reach[20],n,f=0,r=-1,count=0;
void bfs(int v)
{
  int i;
  for(i=1;i<=n;i++)
   if(a[v][i]&&!visited[i])
   {
    visited[i]=1;
    q[++r]=i;
   }
   if(f<=r)
    bfs(q[f++]);
}
void dfs(int v)
{
  int i;
  reach[v]=1;
  for(i=1;i<=n;i++)
   if(a[v][i]&&!reach[i])
   {
    printf("%d->%d\n",v,i);
    count++;
```

```
    dfs(i);
   }
}
int main()
{
  int v,ch,i,j;
  printf("\nenter no. of vertices:");
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  reach[i]=visited[i]=q[i]=0;
  printf("\nEnter graph data in matrix form:\n");
  for(i=1;i<=n;i++)
   for(j=1;j<=n;j++)
    scanf("%d",&a[i][j]);
  printf("\n1.BFS\n2.DFS\n3.Exit\nEnter choice:");
  scanf("%d",&ch);
  switch(ch)
  {
    case 1:printf("\nEnter vertex:");
        scanf("%d",&v);
        bfs(v);
        printf("\nThe nodes that are reacheble from %d are:\n",v);
        for(i=1;i<=n;i++)
         if(visited[i])
        printf("%d ",i);
        break;
    case 2:dfs(1);
        if(count==n-1)
         printf("\ngraph is connected");
        else
         printf("\ngraph is not connected");
```

```
    break;
  case 3:exit(0);
  default:printf("\nInvalid choice");
 }
 return 0;}
```

5. Output: Image of sample output to be attached here



```
input

enter no. of vertices:1

Enter graph data in matrix form:
2

1.BFS
2.DFS
3.Exit
Enter choice:1

Enter vertex:3

The nodes that are reacheble from 3 are:


...Program finished with exit code 0
Press ENTER to exit console.
```

Learning outcomes (What I have learnt):

1. Program to Sort an Array of Integers in Ascending Order Using Heap Sort
2. Syntax and implementation of heap sort.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			