

# Experiment 11: Write a program to get DFS and BFS on a graph using adjacent list.

## 1)DFS

Depth First Search is a graph traversal technique. The source is the first node to be visited, and then we traverse as far as possible from each branch, backtracking when the last node of that branch has been visited.

Adjacency matrix representation: In adjacency matrix representation of a graph, the matrix  $mat[i][j]$  of size  $n \times n$  (where  $n$  is the number of vertices) will represent the edges of the graph where  $mat[i][j] = 1$  represents that there is an edge between the vertices  $i$  and  $j$  while  $mat[i][j] = 0$  represents that there is no edge between the vertices  $i$  and  $j$ .

```
#include<stdio.h>
#include<conio.h>
void DFS(int);
int G[10][10],visited[10],n;
void main()
{
    int i,j;
    printf("Enter number of vertices:");
    scanf("%d",&n);
    printf("\nEnter adjacency matrix of the graph:");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            {scanf("%d",&G[i][j]);}
    for(i=0;i<n;i++)
        visited[i]=0;
    DFS(0);
}
void DFS(int i)
{
    int j;
    printf("\n%d",i);
    visited[i]=1;
    for(j=0;j<n;j++)
        if(!visited[j]&&G[i][j]==1)
            DFS(j);
}
```

}

```
Enter number of vertices:8
Enter adjacency matrix of the graph:0 1 1 1 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 0
0 1 1 0 0 0 0 1
0 0 0 1 1 0 0 1
0 0 0 0 0 1 1 0
0
1
5
2
7
6
3
4
```

## 2)BFS

C program to implement Breadth First Search(BFS). Breadth First Search is an algorithm used to search a Tree or Graph. BFS search starts from the root node then traverses into the next level of graph or tree, if item found it stops otherwise it continues with other nodes in the same level before moving on to the next level.

The purpose of the algorithm is to mark each vertex as visited

The algorithm works as follows:

1. Start by putting any one of the graph's vertices at the back of a queue.
2. Take the front item of the queue and add it to the visited list.

3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the back of the queue.
4. Keep repeating steps 2 and 3 until the queue is empty.

```
#include <stdio.h>
#include <conio.h>
int source,V,E,time,visited[20],G[20][20];
void DFS(int i)
{
    int j;
    visited[i]=1;
    printf(" %d->",i+1);
    for(j=0;j<V;j++)
    {
        if(G[i][j]==1&&visited[j]==0)
            DFS(j);
    }
}
int main()
{
    int i,j,v1,v2;
    printf("Enter the no of edges:");
    scanf("%d",&E);
    printf("Enter the no of vertices:");
    scanf("%d",&V);
    for(i=0;i<V;i++)
    {
        for(j=0;j<V;j++)
            G[i][j]=0;
    }
    for(i=0;i<E;i++)
    {
        printf("Enter the edges");
        scanf("%d%d",&v1,&v2);
        G[v1-1][v2-1]=1;
    }
    for(i=0;i<V;i++)
    {
        for(j=0;j<V;j++)
            printf(" %d ",G[i][j]);
        printf("\n");
    }
    return 0;
}
```

Programiz



Output

```
1 #include <stdio.h>
2 int source,V,E,time,visited[20],G[20][20];
3 void DFS(int i)
4 {
5     int j;
6     visited[i]=1;
7     printf(" %d->",i+1);
8     for(j=0;j<V;j++)
9     {
10         if(G[i][j]==1&&visited[j]==0)
11             DFS(j);
12     }
13 }
14 int main()
15 {
16     int i,j,v1,v2;
17     printf("Enter the no of edges-");
18     scanf("%d",&E);
19     printf("Enter the no of vertices-");
20     scanf("%d",&V);
21     for(i=0;i<V;i++)
```

```
/tmp/M0ixVcxCAc.o
Enter the no of edges:2
Enter the no of vertices:3
Enter the edges1 2
Enter the edges2 3
0 1 0
0 0 1
0 0 0
```

WhatsApp Image...jpeg