|  |  |
| --- | --- |
|  | SVKM’s NMIMS  School of Technology Management & Engineering Navi Mumbai Campus |
| Department of Computer Engineering |

|  |  |
| --- | --- |
| Name: Yash Patil | SAP ID: 70472200140 |
| Semester: III | Year: II |
| Subject: Data Structures and Algorithm | Roll No.: A176 |
| Practical: 9 | Date:16/10/2023 |
| Batch: 1 |  |

Aim: – BFS AND DFS

Code---

/\*

\* C program to implement bfs using adjacency matrix

\*/

#include <stdio.h>

int n, i, j, visited[10], queue[10], front = -1, rear = -1;

int adj[10][10];

void bfs(int v)

{

for (i = 1; i <= n; i++)

if (adj[v][i] && !visited[i])

queue[++rear] = i;

if (front <= rear)

{

visited[queue[front]] = 1;

bfs(queue[front++]);

}

}

void main()

{

int v;

printf("Enter the number of vertices: ");

scanf("%d", &n);

for (i = 1; i <= n; i++)

{

queue[i] = 0;

visited[i] = 0;

}

printf("Enter graph data in matrix form: \n");

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

scanf("%d", &adj[i][j]);

printf("Enter the starting vertex: ");

scanf("%d", &v);

bfs(v);

printf("The node which are reachable are: \n");

for (i = 1; i <= n; i++)

if (visited[i])

printf("%d\t", i);

else

printf("BFS is not possible.

Not all nodes are reachable");

return 0;

}

DFS:

#include<stdio.h>

void DFS(int);

int G[10][10],visited[10],n; //n is no

of vertices and graph is sorted in array

G[10][10]

void main()

{

int i,j;

printf("Enter number of vertices:");

scanf("%d",&n);

//read the adjecency matrix

printf("\nEnter adjecency matrix of the

graph:");

for(i=0;i<n;i++)

for(j=0;j<n;j++)

scanf("%d",&G[i][j]);

//visited is initialized to zero

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);

}