Assignment

Nitesh Kumar

AP19110010542

CSE-‘H’

1.Write a C program depth first search (DFS) using array.

#include <stdio.h>

#include <stdlib.h>

/\* ADJACENCY MATRIX \*/

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("\t\t\tGraphs\n");

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;

}

/\* creating edges :P \*/

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

{

printf("Enter the edges (format: V1 V2) : ");

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");

}

printf("Enter the source: ");

scanf("%d",&source);

DFS(source-1);

return 0;

}

Output

Enter the no of edges:10

Enter the no of vertices:8

Enter the edges (format: V1 V2):1 2

Enter the edges (format: V1 V2):5 2

Enter the edges (format: V1 V2):5 6

Enter the edges (format: V1 V2):5 8

Enter the edges (format: V1 V2):2 5

Enter the edges (format: V1 V2):7 3

Enter the edges (format: V1 V2):8 2

Enter the edges (format: V1 V2):7 8

Enter the edges (format: V1 V2):9 5

Enter the edges (format: V1 V2):3 4

0 1 0 0 0 0 0 0

0 0 0 0 1 0 0 0

0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 0

0 1 0 0 0 1 0 1

0 0 0 0 0 0 0 0

0 0 1 0 0 0 0 1

0 1 0 0 0 0 0 0

Enter the source:2

2-> 5-> 6-> 8->

2.Write a C program breath first search (BFS) using array.

#include<stdio.h>

int a[20][20], q[20], visited[20], n, i, j, f = 0, r = -1;

void bfs(int v) {

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

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

q[++r] = i;

if(f <= r) {

visited[q[f]] = 1;

bfs(q[f++]);

}

}

void main() {

int v;

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

scanf("%d", &n);

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

q[i] = 0;

visited[i] = 0;

}

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

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

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

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

}

}

printf("\n Enter the starting vertex:");

scanf("%d", &v);

bfs(v);

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

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

if(visited[i])

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

else {

printf("\n Bfs is not possible. Not all nodes are reachable");

break;

}

}

}

Output

Enter the number of vertices:4

Enter graph data in matrix form:

5 8 7 2

7 5 8 9

1 4 7 8

3 2 5 4

Enter the starting vertex:4

The node which are reachable are:

1 2 3 4