Md. Asad Chowdhury Dipu

Id: 2019-1-60-093

Section: 1

# Ans#1(Bipartite Graph)

#include<stdio.h>

#include<stdlib.h>

#include<iostream>

#include<queue>

using namespace std;

void PrintGraphMatrix(int G[100][100],int n) {

int i, j;

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

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

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

}

printf("\n");

}

}

void formConnection(int G[100][100],int u,int v){

G[u][v]=1;

G[v][u]=1;

}

bool isBipartite(int G[100][100],int src, int V)

{

int colorArr[V];

for (int i = 0; i < V; ++i)

colorArr[i] = -1;

colorArr[src] = 1;

queue <int> q;

q.push(src);

while (!q.empty())

{

int u = q.front();

q.pop();

if (G[u][u] == 1)

return false;

for (int v = 0; v < V; v++)

{

if (G[u][v] && colorArr[v] == -1)

{

colorArr[v] = 1-colorArr[u];

q.push(v);

}

else if (G[u][v] && colorArr[v] == colorArr[u])

return false;

}

}

return true;

}

int main()

{

int n,edge,u,v,i,j;

int G[100][100];

printf("Enter number of vertex: ");

scanf("%d",n);

printf("Enter number of edges: ");

scanf("%d",&edge);

printf("Enter edges:\n");

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

{

scanf("%d%d",&u,&v);

G[u][v]=1;

G[v][u]=1;

}

printf("\n");

PrintGraphMatrix(G,n);

isBipartite(G,0,n) ? printf("YES, This Graph is Bipartite") : printf("NO, This Graph is Not Bipartite");

return 0;

}