// Topological Sort

#include<stdio.h>

#include<stdlib.h>

int s[100], j, res[100];

void AdjacencyMatrix(int a[][100], int n)

{

int i, j;

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

{

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

{

a[i][j] = 0;

}

}

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

{

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

{

a[i][j] = rand() % 2;

a[j][i] = 0;

}

}

}

void dfs(int u, int n, int a[][100]) { /\* DFS \*/

int v;

s[u] = 1;

for (v = 0; v < n - 1; v++) {

if (a[u][v] == 1 && s[v] == 0) {

dfs(v, n, a);

}

}

j += 1;

res[j] = u;

}

void topological\_order(int n, int a[][100])

{

int i, u;

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

{

s[i] = 0;

}

j = 0;

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

{

if (s[u] == 0) {

dfs(u, n, a);

}

}

return;

}

int main() {

int a[100][100], n, i, j;

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

scanf("%d", &n);

AdjacencyMatrix(a, n);

printf("\tAdjacency Matrix of the graph\n");

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

{

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

{

printf("\t%d", a[i][j]);

}

printf("\n");

}

printf("\nTopological order:\n");

topological\_order(n, a);

for (i = n; i >= 1; i--)

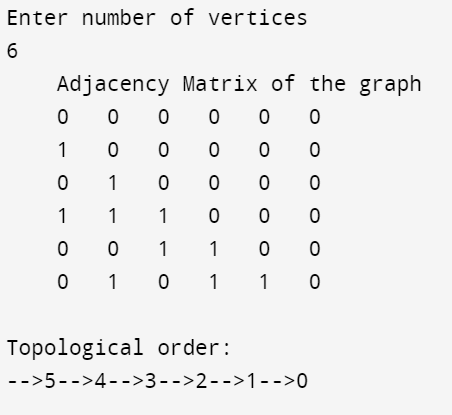
{

printf("-->%d", res[i]);

}

return 0;

}



//Graph colouring

#include <stdbool.h>

#include <stdio.h>

#define V 4

void printSolution(int color[]);

bool isSafe(int v, bool graph[V][V], int color[], int c)

{

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

if (graph[v][i] && c == color[i])

return false;

return true;

}

bool graphColoringUtil(bool graph[V][V], int m, int color[],

int v)

{

if (v == V)

return true;

for (int c = 1; c <= m; c++) {

if (isSafe(v, graph, color, c)) {

color[v] = c;

if (graphColoringUtil(graph, m, color, v + 1)

== true)

return true;

color[v] = 0;

}

}

}

bool graphColoring(bool graph[V][V], int m)

{

int color[V];

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

color[i] = 0;

if (graphColoringUtil(graph, m, color, 0) == false) {

printf("Solution does not exist");

return false;

}

printSolution(color);

return true;

}

void printSolution(int color[])

{

printf("Solution Exists:"

" Following are the assigned colors \n");

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

printf(" %d ", color[i]);

printf("\n");

}

int main()

{

bool graph[V][V] = {

{ 0, 1, 1, 1 },

{ 1, 0, 1, 0 },

{ 1, 1, 0, 1 },

{ 1, 0, 1, 0 },

};

int m = 3;

graphColoring(graph, m);

return 0;

}

