Week 3

#include <stdio.h>

#define MAX 100

int graph[MAX][MAX], inDegree[MAX], vertices;

void calculateInDegree() {

for (int i = 0; i < vertices; i++) {

inDegree[i] = 0;

for (int j = 0; j < vertices; j++) {

if (graph[j][i] == 1) {

inDegree[i]++;

}

}

}

}

void topologicalSort() {

int topologicalOrder[MAX], index = 0;

int count = 0;

calculateInDegree();

while (count < vertices) {

int found = 0;

for (int i = 0; i < vertices; i++) {

if (inDegree[i] == 0) {

topologicalOrder[index++] = i;

inDegree[i] = -1;

found = 1;

count++;

for (int j = 0; j < vertices; j++) {

if (graph[i][j] == 1) {

inDegree[j]--;

}

}

break;

}

}

if (!found) {

printf("DAG detected.\n");

return;

}

}

printf("Topological Sort: ");

for (int i = 0; i < index; i++) {

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

}

printf("\n");

}

int main() {

int edges, u, v;

printf("Enter number of vertices: ");

scanf("%d", &vertices);

printf("Enter number of edges: ");

scanf("%d", &edges);

for (int i = 0; i < vertices; i++) {

for (int j = 0; j < vertices; j++) {

graph[i][j] = 0;

}

}

printf("Enter adjacency matrix:\n");

for (int i = 0; i < vertices; i++) {

for (int j = 0; j < vertices; j++) {

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

}

}

topologicalSort();

return 0;

}

