39. Hamiltanian cycle

#include <stdio.h>

#include <stdbool.h>

#define V 6

bool isSafe(int v, int path[], int graph[V][V], int pos)

{

if (graph[path[pos-1]][v] == 0)

{

return false;

}

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

{

if (path[i] == v)

{

return false;

}

}

return true;

}

bool hamiltonianUtil(int graph[V][V], int path[], int pos)

{

if (pos == V)

{

if (graph[path[pos-1]][path[0]] == 1)

{

return true;

}

else

{

return false;

}

}

for (int v = 1; v < V; v++)

{

if (isSafe(v, path, graph, pos))

{

path[pos] = v;

if (hamiltonianUtil(graph, path, pos + 1) == true)

{

return true;

}

path[pos] = -1;

}

}

return false;

}

bool hamiltonianCycle(int graph[V][V])

{

int path[V];

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

{

path[i] = -1;

}

path[0] = 0;

if (hamiltonianUtil(graph, path, 1) == false)

{

return false;

}

printf("Hamiltonian cycle: ");

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

{

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

}

printf("%d", path[0]);

printf("\n");

return true;

}

int main()

{

int graph[V][V],i,j;

printf("Enter the adjacency matrix for the graph (0 or 1):\n");

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

{

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

{

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

}

}

if (!hamiltonianCycle(graph))

{

printf("No Hamiltonian cycle exists\n");

}

return 0;

}

