
DESIGN AND ANALYSIS OF ALGORITHMS

LAB III SEMESTER

Name : Vardhan Ingole

Section : A4_B3

Roll No : 43

Practical No : 07

Aim : Implement Hamiltonian Cycle using Backtracking.

Task 1 :

The Smart City Transportation Department is designing a night-patrol route for security vehicles.

Each area of the city is represented as a vertex in a graph, and a road between two areas is represented as an edge.

Code :

```
#include <stdio.h>

#define MAX 10

int n;
int G[MAX][MAX];
int x[MAX];
```

```

void NextValue(int k)
{
    int j;

    while (1)
    {
        x[k] = (x[k] + 1) % (n + 1);

        if (x[k] == 0)
            return;

        if (G[x[k - 1]][x[k]] != 0)
        {
            for (j = 1; j < k; j++)
            {
                if (x[j] == x[k])
                    break;
            }

            if (j == k)
            {
                if ((k < n) || (k == n && G[x[n]][x[1]] != 0))
                    return;
            }
        }
    }
}

void Hamiltonian(int k)
{
    int i;

    while (1)
    {
        NextValue(k);

        if (x[k] == 0)
            return;

        if (k == n)
        {
            printf("\nHamiltonian Cycle: ");
            for (i = 1; i <= n; i++)
                printf("%d --> ", x[i]);
            printf("%d", x[1]);
        }
        else
            Hamiltonian(k + 1);
    }
}

```

```

    }
}

int main()
{
    int i, j;
    char ch = 65;

    printf("Enter number of vertices: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix of the graph (%d x %d):\n", n, n);
    for (i = 1; i <= n; i++)
    {
        for (j = i; j <= n; j++)
        {
            if (i != j)
            {
                printf("[%c][%c] = ", ch + i - 1, ch + j - 1);
                scanf("%d", &G[i][j]);
                G[j][i] = G[i][j];
            }
            else
            {
                G[i][j] = 0;
            }
        }
    }

    for (i = 1; i <= n; i++)
        x[i] = 0;

    x[1] = 1;

    printf("\nAll possible Hamiltonian cycles:\n");
    Hamiltonian(2);

    return 0;
}

```

Test Case 1:

```
interpreter=ml
● Enter number of vertices: 5
Enter adjacency matrix of the graph (5 x 5):
[A][B] = 1
[A][C] = 1
[A][D] = 0
[A][E] = 1
[B][C] = 1
[B][D] = 1
[B][E] = 0
[C][D] = 1
[C][E] = 0
[D][E] = 1

All possible Hamiltonian cycles:

Hamiltonian Cycle: 1 --> 2 --> 3 --> 4 --> 5 --> 1
Hamiltonian Cycle: 1 --> 3 --> 2 --> 4 --> 5 --> 1
Hamiltonian Cycle: 1 --> 5 --> 4 --> 2 --> 3 --> 1
Hamiltonian Cycle: 1 --> 5 --> 4 --> 3 --> 2 --> 1
○ PS D:\Sem_III\DAA_Laboratory>
```

Test Case 2:

● Enter number of vertices: 5
Enter adjacency matrix of the graph (5 x 5):

```
[A][B] = 1  
[A][C] = 1  
[A][D] = 0  
[A][E] = 1  
[B][C] = 1  
[B][D] = 1  
[B][E] = 0  
[C][D] = 1  
[C][E] = 1  
[D][E] = 1
```

All possible Hamiltonian cycles:

```
Hamiltonian Cycle: 1 --> 2 --> 3 --> 4 --> 5 --> 1  
Hamiltonian Cycle: 1 --> 2 --> 4 --> 3 --> 5 --> 1  
Hamiltonian Cycle: 1 --> 2 --> 4 --> 5 --> 3 --> 1  
Hamiltonian Cycle: 1 --> 3 --> 2 --> 4 --> 5 --> 1  
Hamiltonian Cycle: 1 --> 3 --> 5 --> 4 --> 2 --> 1  
Hamiltonian Cycle: 1 --> 5 --> 3 --> 4 --> 2 --> 1  
Hamiltonian Cycle: 1 --> 5 --> 4 --> 2 --> 3 --> 1  
Hamiltonian Cycle: 1 --> 5 --> 4 --> 3 --> 2 --> 1
```

○ PS D:\Sem_III\DAA_Laboratory> █

Hamiltonian Path



Difficulty: Medium

Accuracy: 40.8%

Submissions: 46K+

Points: 4

Given an undirected graph with n vertices and m edges, your task is to determine if a Hamiltonian path exists in the graph.

A **Hamiltonian path** is a path in an undirected graph that visits each vertex exactly once.

You are provided the following:

- **n**: The number of vertices in the graph.
- **m**: The number of edges in the graph.
- **edges[][]**: A 2D list where each element `edges[i]` represents an edge between two vertices `edges[i][0]` and `edges[i][1]`.

Examples:

Input: $n = 4, m = 4$

`edges[][] = { {1,2}, {2,3}, {3,4}, {2,4} }`

Output: 1

Explanation: There is a hamiltonian path: 1 -> 2 -> 3 -> 4

Input: $n = 4, m = 3$

`edges[][] = { {1,2}, {2,3}, {2,4} }`

Output: 0

Explanation: It can be proved that there is no hamiltonian path in the given graph.

Constraints:

$1 \leq n \leq 10$

$1 \leq m \leq 15$


Size of `edges[i]` is 2



$1 \leq \text{edges}[i][0], \text{edges}[i][1] \leq n$

```
Python3 Start Timer
1 class Solution:
2     def solve(self, u, count, visited, n, adj):
3         visited[u] = True
4
5         if count == n:
6             return True
7
8         for v in adj[u]:
9             if not visited[v]:
10                if self.solve(v, count + 1, visited, n, adj):
11                    return True
12
13        visited[u] = False
14        return False
15
16    def check(self, n, m, edges):
17        adj = [[] for _ in range(n + 1)]
18        for u, v in edges:
19            adj[u].append(v)
20            adj[v].append(u)
21
22        for i in range(1, n + 1):
23            visited = [False] * (n + 1)
24            if self.solve(i, 1, visited, n, adj):
25                return 1
26
27        return 0
```

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully  [Suggest Feedback](#)

Test Cases Passed 52 / 52	Attempts : Correct / Total 1 / 1 Accuracy : 100%
Points Scored  4 / 4 Your Total Score: 12 	Time Taken 0.03

Link :

<https://www.geeksforgeeks.org/problems/hamiltonian-path2522/1>