


# Practical No 7 : Competitive Coding


Name- Anjali Lanjewar

Section- A4-B2

Roll no- 25



Courses ▼ Tutorial

 [Problem](#) [Editorial](#) [Submissions](#) [Comments](#)

## 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

C++ (17) Start Timer

```
1 class Solution {
2     public:
3     virtual bool dfs(int i, int n, vector<vector<int>> &adj, vector<bool> &visited, vector<int> &store){
4         visited[i] = true;
5         store.push_back(i);
6
7         if(store.size() == n)
8             return true;
9
10        for(int x: adj[i]){
11            if(!visited[x]){
12                if(dfs(x, n, adj, visited, store))
13                    return true;
14            }
15        }
16
17        visited[i] = false;
18        store.pop_back();
19        return false;
20    }
21
22    virtual bool check(int n, int m, vector<vector<int>> edges) {
23
24        vector<vector<int>> adj(n+1);
25        for(auto &v: edges){
26            adj[v[0]].push_back(v[1]);
27            adj[v[1]].push_back(v[0]);
28        }
29
30        for(int i=1;i<=n;i++){
31            vector<bool> visited(n+1, false);
32            vector<int> store;
33
34            if(dfs(i, n, adj, visited, store))
35                return true;
36        }
37        return false;
38    }
39};
```

Output Window

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

## Compilation Completed

Case 1

Input: 

```
4 4
1 2 2 3 3 4 2 4
```

Your Output:

1

Expected Output:

1

## Problem Solved Successfully ✓

[Suggest Feedback](#)

Test Cases Passed

**52 / 52**

Attempts : Correct / Total

**1 / 2**

Accuracy : 50%

Points Scored ⓘ

**4 / 4**

Your Total Score: 12 ↑

Time Taken

**0.01**