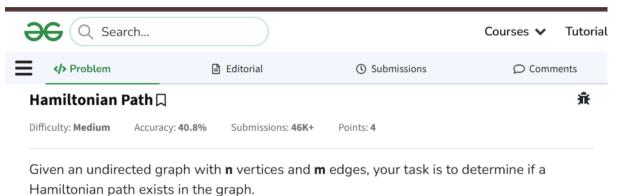
## **Practical No 7**: Competitive Coding

Name- Anjali Lanjewar Section- A4-B2 Roll no- 25



A <u>Hamiltonian path</u> 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 \rightarrow 2 \rightarrow 3 \rightarrow 4
```

```
C++ (17)
                        1 - class Solution {
2
     public:
       virtual bool dfs(int i, int n, vector<vector<int>> &adj, vector<bool> &visited, vector<int> &store){
3 +
4
           visited[i] = true;
           store.push_back(i);
5
6
           if(store.size() == n)
8
            return true;
9
.0 -
            for(int x: adj[i]){
1 -
                if(!visited[x]){
                   if(dfs(x, n, adj, visited, store))
return true;
.2
.3
.4
                }
            }
.6
           visited[i] = false;
.7
8.
            store.pop_back();
.9
           return false;
10
1
2
!3 ₹
        virtual bool check(int n, int m, vector<vector<int>> edges) {
4
.5
          vector<vector<int>> adj(n+1);
6 -
          for(auto &v: edges){
.7
            adj[v[0]].push_back(v[1]);
           adj[v[1]].push_back(v[0]);
8
9
0
         for(int i=1;i<=n;i++){
    vector<bool> visited(n+1, false);
1 -
2
             vector<int> store;
3
4
15
              if(dfs(i, n, adj, visited, store))
6
                return true;
7
8
9
          return false;
0
2
   };
  Output Window
 Compilation Results
                        Custom Input
                                         Y.O.G.I. (AI Bot)
  Compilation Completed
    • Case 1
    Input: 🗘
     44
    12233424
    Your Output:
```

Expected Output:

