Name : Mohammad Anas Ajaz

Roll No. 61

Batch : A4(B4)

Practical No 7

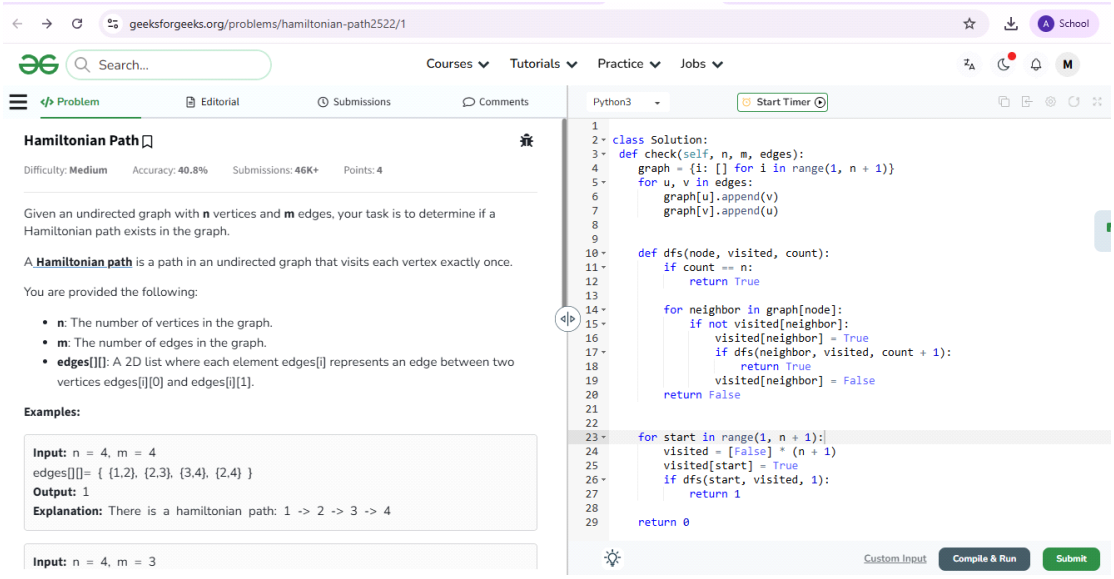
Competitive Coding Link:

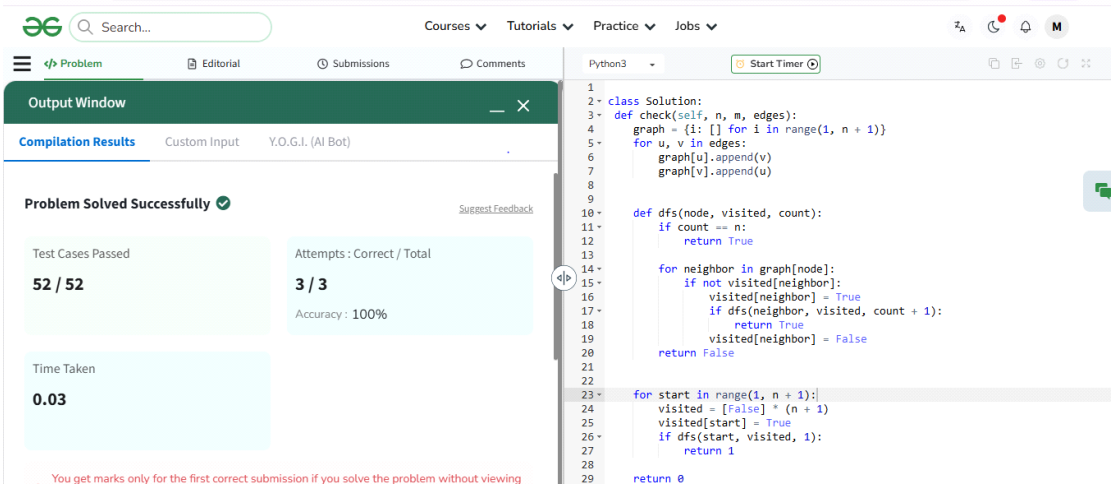
https://www.geeksforgeeks.org/problems/hamiltonian-path2522/1

code :

```
class Solution:
 def check(self, n, m, edges):
   graph = {i: [] for i in range(1, n + 1)}
   for u, v in edges:
      graph[u].append(v)
      graph[v].append(u)



   def dfs(node, visited, count):
      if count == n:
         return True


      for neighbor in graph[node]:
         if not visited[neighbor]:
            visited[neighbor] = True
            if dfs(neighbor, visited, count + 1):
               return True
            visited[neighbor] = False
```

```
    return False


for start in range(1, n + 1):
    visited = [False] * (n + 1)
    visited[start] = True
    if dfs(start, visited, 1):
        return 1

return 0
```




Search...
Courses
Tutorials
Practice
Jobs
Problem
Editorial
Submissions
Comments
Python3
Start Timer
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
3 / 3
Accuracy : 100%
Time Taken
0.03
You get marks only for the first correct submission if you solve the problem without viewing
class Solution:
def check(self, n, m, edges):
graph = {i: [] for i in range(1, n + 1)}
for u, v in edges:
graph[u].append(v)
graph[v].append(u)
def dfs(node, visited, count):
if count == n:
return True
for neighbor in graph[node]:
if not visited[neighbor]:
visited[neighbor] = True
if dfs(neighbor, visited, count + 1):
return True
visited[neighbor] = False
return False
for start in range(1, n + 1):
visited = [False] * (n + 1)
visited[start] = True
if dfs(start, visited, 1):
return 1
return 0