

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 #include <algorithm>
6 using namespace std;
7
8 class AirportNode {
9 public:
10     string airport;
11     vector<string> connections;
12     bool isReachable;
13     vector<string> unreachableConnections;
14
15     AirportNode(string airport) {
16         this->airport = airport;
17         connections = {};
18         isReachable = true;
19         unreachableConnections = {};
20     }
21 };
22
23 unordered_map<string, AirportNode*>
24 createAirportGraph(vector<string> airports, vector<vector<string>>> roi
25 vector<AirportNode*>
26 getUnreachableAirportNodes(unordered_map<string, AirportNode*> airpor
27     vector<string> airports, string startingAir
28 void depthFirstTraverseAirports(
29     unordered_map<string, AirportNode*> airportGraph, string airport,
30     unordered_map<string, bool> *visitedAirports);
31 void markUnreachableConnections(
32     unordered_map<string, AirportNode*> airportGraph,
33     vector<AirportNode*> unreachableAirportNodes);
```

Solution 1Solution 2Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 int airportConnections(vector<string> airports, vector<vector<string>>
5     string startingAirport) {
6     // Write your code here.
7     return -1;
8 }
9
```

Our Tests

Custom Output

Submit Code

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.

2

3 #include <vector>

4 #include <unordered\_map>

5 #include <algorithm>

6 using namespace std;

7

8 class AirportNode {

9 public:

10 string airport;

11 vector<string> connections;

12 bool isReachable;

13 vector<string> unreachableConnections;

14

15 AirportNode(string airport) {

16 this->airport = airport;

17 connections = {};

18 isReachable = true;

19 unreachableConnections = {};

20 }

21 }

22

23 unordered\_map<string, AirportNode\*>

24 createAirportGraph(vector<string> airports, vector<vector<string>>> roi

25 vector<AirportNode\*>

26 getUnreachableAirportNodes(unordered\_map<string, AirportNode\*> airpor

27 vector<string> airports, string startingAir

28 void depthFirstTraverseAirports(

29 unordered\_map<string, AirportNode\*> airportGraph, string airport,

30 unordered\_map<string, bool> \*visitedAirports);

31 void markUnreachableConnections(

32 unordered\_map<string, AirportNode\*> airportGraph,

33 vector<AirportNode\*> unreachableAirportNodes);

1 #include <vector>

2 using namespace std;

3

4 int airportConnections(vector<string> airports, vector<vector<string>>

5 string startingAirport) {

6 // Write your code here.

7 return -1;

8 }

9

