Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {

  // O(a * (a + r) + a + r + alog(a)) time | O(a + r) space - where a
  // r is the number of routes
  public static int airportConnections(
      List<String> airports, List<List<String>> routes, String startir
    Map<String, AirportNode> airportGraph = createAirportGraph(airport
    List<AirportNode> unreachableAirportNodes =
        getUnreachableAirportNodes(airportGraph, airports, startingAir
    markUnreachableConnections(airportGraph, unreachableAirportNodes);
    return getMinNumberOfNewConnections(airportGraph, unreachableAirpo
  }

  // O(a + r) time | O(a + r) space
  public static Map<String, AirportNode> createAirportGraph(
      List<String> airports, List<List<String>> routes) {
    Map<String, AirportNode> airportGraph = new HashMap<String, Airpor
    for (String airport : airports) {
      airportGraph.put(airport, new AirportNode(airport));
    }
    for (List<String> route : routes) {
      String airport = route.get(0);
      String connection = route.get(1);
      airportGraph.get(airport).connections.add(connection);
    }
    return airportGraph;
  }

  // O(a + r) time | O(a) space
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static int airportConnections(
      List<String> airports, List<List<String>> routes, String startin
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.