

Our Solution(s)	Run Code	Your Solutions	Run Code
-----------------	----------	----------------	----------

Solution 1	Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 import java.util.*; 4 5 class Program { 6 7 // O(a * (a + r) + a + r + alog(a)) time O(a + r) space - where a 8 // r is the number of routes 9 public static int airportConnections(10 List<String> airports, List<List<String>> routes, String startingAirport) { 11 Map<String, AirportNode> airportGraph = createAirportGraph(airports, routes); 12 List<AirportNode> unreachableAirportNodes = 13 getUnreachableAirportNodes(airportGraph, airports, startingAirport); 14 markUnreachableConnections(airportGraph, unreachableAirportNodes); 15 return getMinNumberOfNewConnections(airportGraph, unreachableAirportNodes); 16 } 17 18 // O(a + r) time O(a + r) space 19 public static Map<String, AirportNode> createAirportGraph(20 List<String> airports, List<List<String>> routes) { 21 Map<String, AirportNode> airportGraph = new HashMap<String, AirportNode>(); 22 for (String airport : airports) { 23 airportGraph.put(airport, new AirportNode(airport)); 24 } 25 for (List<String> route : routes) { 26 String airport = route.get(0); 27 String connection = route.get(1); 28 airportGraph.get(airport).connections.add(connection); 29 } 30 return airportGraph; 31 } 32 33 // O(a + r) time O(a) space</pre>	<pre>1 import java.util.*; 2 3 class Program { 4 public static int airportConnections(5 List<String> airports, List<List<String>> routes, String startingAirport) { 6 // Write your code here. 7 return -1; 8 } 9 } 10</pre>		

Our Tests	Custom Output	Submit Code
-----------	---------------	-------------

<pre>1 import java.util.*; 2 3 class Program { 4 public static int airportConnections(5 List<String> airports, List<List<String>> routes, String startingAirport) { 6 // Write your code here. 7 return -1; 8 } 9 }</pre>		
--	--	--

```

14         return result;
15     }
16     return result;
17 }
18
19 String toString() {
20     return "100";
21 }
22
23 @Test
24 public void testToString() {
25     assertEquals("toString() returns 100", "100", toString());
26     assertEquals("toString() returns 100", "100", toString());
27 }
28
29 @Test
30 public void testToString() {
31     assertEquals("toString() returns 100", "100", toString());
32     assertEquals("toString() returns 100", "100", toString());
33 }

```

Run or submit code when you're ready.