

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(a * (a + r) + (a + r) + alog(a)) time | O(a + r) space
5     func airportConnections(_ airports: [String], _ routes: [[String]], _ startingAirport: String) -> Int {
6         var airportGraph = createAirportGraph(airports, routes)
7         var unreachableAirportNodes = getUnreachableAirportNodes(airports, &airportGraph, startingAirport)
8         addChildrenToUnreachableAirportNodes(airportGraph, unreachableAirportNodes)
9
10        return getMinimumNumberOfNewConnections(&airportGraph, &unreachableAirportNodes)
11    }
12
13    // O(a + r) time | O(a + r) space
14    func createAirportGraph(_ airports: [String], _ routes: [[String]]) -> [String: AirportNode] {
15        var airportGraph = [String: AirportNode]()
16
17        for airportCode in airports {
18            airportGraph[airportCode] = AirportNode(airportCode)
19        }
20
21        for route in routes {
22            let origin = route[0]
23            let destination = route[1]
24
25            if let airportNode = airportGraph[origin] {
26                airportNode.directConnections.append(destination)
27                airportGraph[origin] = airportNode
28            }
29        }
30
31        return airportGraph
32    }
33
34    // O(a + r) time | O(a) space
35    func getUnreachableAirportNodes(_ airports: [String], _ airportsGraph: inout [String: AirportNode], _ startingAirport: String) -> [AirportNode] {
36        var visitedAirports = [String: Bool]()
37        depthFirstTraverseAirports(airportsGraph, startingAirport, &visitedAirports)
38
39        var unreachableAirportNodes = [AirportNode]()
40        for airportCode in airports {
41            if visitedAirports.keys.contains(airportCode) {
42                continue
43            }
44
45            if let airportNode = airportsGraph[airportCode] {
46                airportNode.isReachable = false
47                airportsGraph[airportCode] = airportNode
48                unreachableAirportNodes.append(airportNode)
49            }
50        }
51
52        return unreachableAirportNodes
53    }
54
55    func depthFirstTraverseAirports(_ airportsGraph: [String: AirportNode], _ airport: String, _ visitedAirports: inout [String: Bool]) {
56        if visitedAirports.keys.contains(airport) {
57            return
58        }
59
60        visitedAirports[airport] = true
61
62        if let airportNode = airportsGraph[airport] {
63            let directConnections = airportNode.directConnections
64
65            for connection in directConnections {
66                depthFirstTraverseAirports(airportsGraph, connection, &visitedAirports)
67            }
68        }
69    }
70
71    // O(a * (a + r)) time | O(a) space
72    func addChildrenToUnreachableAirportNodes(_ airportsGraph: [String: AirportNode], _ unreachableAirportNodes: [AirportNode]) {
73        for airportNode in unreachableAirportNodes {
74            var visitedAirports = [String: Bool]()
75            var childConnections = [String]()
76            let airportCode = airportNode.airportCode
77
78            depthFirstAddChildConnections(airportCode, airportsGraph, &visitedAirports, &childConnections)
79            airportNode.allChildConnections = childConnections
80        }
81    }
82
83    func depthFirstAddChildConnections(_ airportCode: String, _ airportsGraph: [String: AirportNode], _ visitedAirports: inout [String: Bool], _ childConnections: inout [String]) {
84        if visitedAirports.keys.contains(airportCode) {
85            return
86        }
87
88        if let airportNode = airportsGraph[airportCode], airportNode.isReachable {
89            return
90        }
91
92        visitedAirports[airportCode] = true
93        childConnections.append(airportCode)
94
95        if let airportNode = airportsGraph[airportCode] {
96            let directConnections = airportNode.directConnections
97
98            for connection in directConnections {
99                depthFirstAddChildConnections(connection, airportsGraph, &visitedAirports, &childConnections)
100            }
101        }
102    }
103
104    // O(alog(a) + a + r) time | O(1) space
105    func getMinimumNumberOfNewConnections(_ airportGraph: inout [String: AirportNode], _ unreachableAirportNodes: inout [AirportNode]) -> Int {
106        var numberOfNewConnections = 0
107        unreachableAirportNodes = unreachableAirportNodes.sorted(by: { $0.allChildConnections.count > $1.allChildConnections.count })
108
109        for airportNode in unreachableAirportNodes {
110            if airportNode.isReachable {
111                continue
112            }
113
114            numberOfNewConnections += 1
115        }
```

```
116     for child in airportNode.allChildConnections {
117         if let airportNode = airportGraph[child] {
```