

AlgoExpert

Quad Layout

Swift

12px

Sublime

Monokai

00:00:00

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4 // O(br) time | O(br) space - where b is the number of blocks and r is the number of requirements
5 func apartmentHunting(_ blocks: [[String: Bool]], _ requirements: [String]) -> Int {
6 let minDistancesFromBlocks = requirements.map { getMinDistances(blocks, \$0) }
7 let maxDistancesAtBlocks = getMaxDistancesAtBlocks(blocks, minDistancesFromBlocks)
8
9 return getIndexAtMinValue(maxDistancesAtBlocks)
10 }
11
12 func getMinDistances(_ blocks: [[String: Bool]], _ requirement: String) -> [Int] {
13 var minDistances = Array(repeating: -1, count: blocks.count)
14 var closestRequirementIndex = Int.max
15
16 for i in 0 ..< blocks.count {
17 if let requirementAvailable = blocks[i][requirement], requirementAvailable {
18 closestRequirementIndex = i
19 }
20
21 minDistances[i] = distanceBetween(i, closestRequirementIndex)
22 }
23
24 for i in (0 ..< blocks.count).reversed() {
25 if let requirementAvailable = blocks[i][requirement], requirementAvailable {
26 closestRequirementIndex = i
27 }
28
29 minDistances[i] = min(minDistances[i], distanceBetween(i, closestRequirementIndex))
30 }
31
32 return minDistances
33 }
34
35 func getMaxDistancesAtBlocks(_ blocks: [[String: Bool]], _ minDistancesFromBlocks: [[Int]]) -> [Int] {
36 var maxDistancesAtBlocks = Array(repeating: -1, count: blocks.count)
37
38 for i in 0 ..< blocks.count {
39 let minDistancesAtBlock = minDistancesFromBlocks.map { \$0[i] }
40
41 if let max = minDistancesAtBlock.max() {
42 maxDistancesAtBlocks[i] = max
43 }
44 }
45
46 return maxDistancesAtBlocks
47 }
48
49 func getIndexAtMinValue(_ array: [Int]) -> Int {
50 var indexAtMinValue = 0
51 var minValue = Int.max
52
53 for i in 0 ..< array.count {
54 let currentValue = array[i]
55
56 if currentValue < minValue {
57 minValue = currentValue
58 indexAtMinValue = i
59 }
60 }
61
62 return indexAtMinValue
63 }
64
65 func distanceBetween(_ a: Int, _ b: Int) -> Int {
66 return abs(a - b)
67 }
68 }
69

