

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 class Program { 4 // O(b^2 * r) time O(b) space 5 func apartmentHunting(_ blocks: [[String: Bool]], _ requirements: [String: Int]) -> Int { 6 var maxDistancesAtBlocks = Array(repeating: -Int.max, count: blocks.count) 7 for i in 0 ..< blocks.count { 8 for requirement in requirements { 9 var closestReqDistance = Int.max 10 11 for j in 0 ..< blocks.count { 12 if let requirementAvailable = blocks[j][requirement], requirementAvailable { 13 closestReqDistance = min(closestReqDistance, distanceBetween(i, j, requirement)) 14 } 15 } 16 17 maxDistancesAtBlocks[i] = max(maxDistancesAtBlocks[i], closestReqDistance) 18 } 19 } 20 21 return getIndexAtMinValue(maxDistancesAtBlocks) 22 } 23 24 func getIndexAtMinValue(_ array: [Int]) -> Int { 25 var indexAtMinValue = 0 26 var minValue = Int.max 27 28 for i in 0 ..< array.count { 29 let currentValue = array[i] 30 31 if currentValue < minValue { 32 minValue = currentValue 33 indexAtMinValue = i 34 } 35 } 36 37 return indexAtMinValue 38 } 39 }</pre>		<pre>1 class Program { 2 func apartmentHunting(_ blocks: [[String: Bool]], _ requirements: [String: Int]) -> Int { 3 // Write your code here. 4 return -1 5 } 6 } 7</pre>

Our Tests

Custom Output

Submit Code

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

2

3 class Program {

4 // O(b^2 * r) time | O(b) space

5 func apartmentHunting(_ blocks: [[String: Bool]], _ requirements: [String: Int]) -> Int {

6 var maxDistancesAtBlocks = Array(repeating: -Int.max, count: blocks.count)

7 for i in 0 ..< blocks.count {

8 for requirement in requirements {

9 var closestReqDistance = Int.max

10

11 for j in 0 ..< blocks.count {

12 if let requirementAvailable = blocks[j][requirement], requirementAvailable {

13 closestReqDistance = min(closestReqDistance, distanceBetween(i, j, requirement))

14 }

15 }

16 maxDistancesAtBlocks[i] = max(maxDistancesAtBlocks[i], closestReqDistance)

17 }

18 }

19 }

20

21 return getIndexAtMinValue(maxDistancesAtBlocks)

22 }

23

24 func getIndexAtMinValue(_ array: [Int]) -> Int {

25 var indexAtMinValue = 0

26 var minValue = Int.max

27

28 for i in 0 ..< array.count {

29 let currentValue = array[i]

30

31 if currentValue < minValue {

32 minValue = currentValue

33 indexAtMinValue = i

34 }

35 }

36

37 return indexAtMinValue

38 }

39 }

1

2

3

4

5

6

7

Run or submit code when you're ready.