AlgoExpert Quad Layout Go 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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

package main

import "math"

type Block map[string]bool

// O(b^2*r) time | O(b) space - where b is the number of blocks
// and r is the number of requirements.
func ApartmentHunting(blocks []Block, reqs []string) int {
  maxDistancesAtBlocks := make([]int, len(blocks))
  for i := range blocks {
    maxDistancesAtBlocks[i] = -1
    for _, req := range reqs {
      closestReqDistance := math.MaxInt32
      for j := range blocks {
        if blocks[j][req] {
          closestReqDistance = min(closestReqDistance, distanceBetween(i, j))
        }
      }
      maxDistancesAtBlocks[i] = max(maxDistancesAtBlocks[i], closestReqDistance)
    }
  }

  var optimalBlockIdx int
  smallestMaxDistance := math.MaxInt32
  for i, currentDistance := range maxDistancesAtBlocks {
    if currentDistance < smallestMaxDistance {
      smallestMaxDistance = currentDistance
      optimalBlockIdx = i
    }
  }
  return optimalBlockIdx
}

func distanceBetween(a, b int) int {
  if a > b {
    return a - b
  }
  return b - a
}

func min(a, b int) int {
  if a < b {
    return a
  }
  return b
}

func max(a, b int) int {
  if a > b {
    return a
  }
  return b
}
```