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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(br) time | O(br) space - where b is the number of blocks and r is the number of requirements
def apartmentHunting(blocks, reqs):
    minDistancesFromBlocks = list(map(lambda req: getMinDistances(blocks, req), reqs))
    maxDistancesAtBlocks = getMaxDistancesAtBlocks(blocks, minDistancesFromBlocks)
    return getIdxAtMinValue(maxDistancesAtBlocks)


def getMinDistances(blocks, req):
    minDistances = [0 for block in blocks]
    closestReqIdx = float("inf")
    for i in range(len(blocks)):
        if blocks[i][req]:
            closestReqIdx = i
        minDistances[i] = distanceBetween(i, closestReqIdx)
    for i in reversed(range(len(blocks))):
        if blocks[i][req]:
            closestReqIdx = i
        minDistances[i] = min(minDistances[i], distanceBetween(i, closestReqIdx))
    return minDistances


def getMaxDistancesAtBlocks(blocks, minDistancesFromBlocks):
    maxDistancesAtBlocks = [0 for block in blocks]
    for i in range(len(blocks)):
        minDistancesAtBlock = list(map(lambda distances: distances[i], minDistancesFromBlocks))
        maxDistancesAtBlocks[i] = max(minDistancesAtBlock)
    return maxDistancesAtBlocks


def getIdxAtMinValue(array):
    idxAtMinValue = 0
    minValue = float("inf")
    for i in range(len(array)):
        currentValue = array[i]
        if currentValue < minValue:
            minValue = currentValue
            idxAtMinValue = i
    return idxAtMinValue


def distanceBetween(a, b):
    return abs(a - b)

```