AlgoExpert Quad Layout C# 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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

using System;
using System.Collections.Generic;

public class Program {
  // O(b^2*r) time | O(b) space - where b is the number of blocks and r is the number of requirements
  public static int ApartmentHunting(List<Dictionary<string, bool> > blocks, string[] reqs) {
    int[] maxDistancesAtBlocks = new int[blocks.Count];
    Array.Fill(maxDistancesAtBlocks, Int32.MinValue);

    for (int i = 0; i < blocks.Count; i++) {
      foreach (string req in reqs) {
        int closestReqDistance = Int32.MaxValue;
        for (int j = 0; j < blocks.Count; j++) {
          if (blocks[j][req]) {
            closestReqDistance = Math.Min(closestReqDistance, distanceBetween(
                  i,
                  j));
          }
        }
        maxDistancesAtBlocks[i] = Math.Max(maxDistancesAtBlocks[i],
            closestReqDistance);
      }
    }
    return getIdxAtMinValue(maxDistancesAtBlocks);
  }

  public static int getIdxAtMinValue(int[] array) {
    int idxAtMinValue = 0;
    int minValue = Int32.MaxValue;
    for (int i = 0; i < array.Length; i++) {
      int currentValue = array[i];
      if (currentValue < minValue) {
        minValue = currentValue;
        idxAtMinValue = i;
      }
    }
    return idxAtMinValue;
  }

  public static int distanceBetween(int a, int b) {
    return Math.Abs(a - b);
  }
}

```