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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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

import java.util.*;

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<Map<String, Boolean>> blocks, String[] reqs) {
    int[] maxDistancesAtBlocks = new int[blocks.size()];
    Arrays.fill(maxDistancesAtBlocks, Integer.MIN_VALUE);

    for (int i = 0; i < blocks.size(); i++) {
      for (String req : reqs) {
        int closestReqDistance = Integer.MAX_VALUE;
        for (int j = 0; j < blocks.size(); j++) {
          if (blocks.get(j).get(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 = Integer.MAX_VALUE;
    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);
  }
}
```