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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 2
 Solution 1
 1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
    import java.util.*;
    class Program {
     // O(br) time | O(br) 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[][] minDistancesFromBlocks = new int[reqs.length][];
10
        for (int i = 0; i < reqs.length; i++) {
11
          minDistancesFromBlocks[i] = getMinDistances(blocks, reqs[i]);
12
        int[] maxDistancesAtBlocks = getMaxDistancesAtBlocks(blocks, minDistancesFromBlocks);
13
14
        return getIdxAtMinValue(maxDistancesAtBlocks);
15
16
17
      public static int[] getMinDistances(List<Map<String, Boolean>> blocks, String req) {
18
        int[] minDistances = new int[blocks.size()];
19
        int closestReqIdx = Integer.MAX_VALUE;
        for (int i = 0; i < blocks.size(); i++) {</pre>
20
21
          if (blocks.get(i).get(req)) closestReqIdx = i;
22
          minDistances[i] = distanceBetween(i, closestReqIdx);
23
24
        for (int i = blocks.size() - 1; i >= 0; i--) {
25
          if (blocks.get(i).get(req)) closestReqIdx = i;
26
          minDistances[i] = Math.min(minDistances[i], distanceBetween(i, closestReqIdx));
27
28
        return minDistances;
29
30
31
      public static int[] getMaxDistancesAtBlocks(
32
          List<Map<String, Boolean>> blocks, int[][] minDistancesFromBlocks) {
        int[] maxDistancesAtBlocks = new int[blocks.size()];
33
34
        for (int i = 0; i < blocks.size(); i++) {</pre>
35
          int[] minDistancesAtBlock = new int[minDistancesFromBlocks.length];
          for (int j = 0; j < minDistancesFromBlocks.length; j++) {</pre>
36
            minDistancesAtBlock[j] = minDistancesFromBlocks[j][i];
37
38
          maxDistancesAtBlocks[i] = arrayMax(minDistancesAtBlock);
39
40
41
        return maxDistancesAtBlocks;
42
43
44
      public static int getIdxAtMinValue(int[] array) {
        int idxAtMinValue = 0;
45
46
        int minValue = Integer.MAX_VALUE;
47
        for (int i = 0; i < array.length; i++) {</pre>
48
          int currentValue = array[i];
49
          if (currentValue < minValue) {</pre>
            minValue = currentValue;
50
51
            idxAtMinValue = i;
52
53
54
        return idxAtMinValue;
55
56
57
      public static int distanceBetween(int a, int b) {
58
        return Math.abs(a - b);
59
60
```

61

62

63

64

65

66 67 68

69 70 }

public static int arrayMax(int[] array) {

int max = array[0];

if (a > max) {

max = a;

return max;

 $\quad \text{for (int a : array) } \{$