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

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
_{\rm 1} \, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
    // O(br) time \mid O(br) space - where b is the number of blocks and r is the number of requirements
    function apartmentHunting(blocks, reqs) {
      const minDistancesFromBlocks = reqs.map(req => getMinDistances(blocks, req));
      const maxDistancesAtBlocks = getMaxDistancesAtBlocks(blocks, minDistancesFromBlocks);
      return getIdxAtMinValue(maxDistancesAtBlocks);
 8 }
10 function getMinDistances(blocks, req) {
      const minDistances = new Array(blocks.length);
12
      let closestReqIdx = Infinity;
      for (let i = 0; i < blocks.length; i++) {</pre>
13
        if (blocks[i][req]) closestReqIdx = i;
14
        minDistances[i] = distanceBetween(i, closestReqIdx);
15
16
      for (let i = blocks.length - 1; i >= 0; i--) {
17
18
        if (blocks[i][req]) closestReqIdx = i;
19
        minDistances[i] = Math.min(minDistances[i], distanceBetween(i, closestReqIdx));
20
21
      return minDistances;
22 }
23
24
    function \ getMaxDistances AtBlocks (blocks, \ minDistances FromBlocks) \ \{
      const maxDistancesAtBlocks = new Array(blocks.length);
26
      for (let i = 0; i < blocks.length; i++) {</pre>
27
        const minDistancesAtBlock = minDistancesFromBlocks.map(distances => distances[i]);
28
        maxDistancesAtBlocks[i] = Math.max(...minDistancesAtBlock);
29
30
      return maxDistancesAtBlocks;
31 }
33 function getIdxAtMinValue(array) {
34
     let idxAtMinValue = 0;
      let minValue = Infinity;
35
      for (let i = 0; i < array.length; i++) \{
36
37
        const currentValue = array[i];
38
        if (currentValue < minValue) {</pre>
39
          minValue = currentValue;
40
          idxAtMinValue = i;
41
42
43
      return idxAtMinValue;
44 }
45
    function distanceBetween(a, b) {
47
      return Math.abs(a - b);
48 }
49
50 exports.apartmentHunting = apartmentHunting;
```

Solution 1 Solution 2