Our Solution(s) Run Code

Solution 1 Solution 2

```java
// 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
  // requirements
  public static int apartmentHunting(List<Map<String, Boolean>> block
    int[][] minDistancesFromBlocks = new int[reqs.length][];
    for (int i = 0; i < reqs.length; i++) {
      minDistancesFromBlocks[i] = getMinDistances(blocks, reqs[i]);
    }
    int[] maxDistancesAtBlocks = getMaxDistancesAtBlocks(blocks, minDi
    return getIdxAtMinValue(maxDistancesAtBlocks);
  }

  public static int[] getMinDistances(List<Map<String, Boolean>> block
    int[] minDistances = new int[blocks.size()];
    int closestReqIdx = Integer.MAX_VALUE;
    for (int i = 0; i < blocks.size(); i++) {
      if (blocks.get(i).get(req)) closestReqIdx = i;
      minDistances[i] = distanceBetween(i, closestReqIdx);
    }
    for (int i = blocks.size() - 1; i >= 0; i--) {
      if (blocks.get(i).get(req)) closestReqIdx = i;
      minDistances[i] = Math.min(minDistances[i], distanceBetween(i, c
    }
    return minDistances;
  }

  public static int[] getMaxDistancesAtBlocks(
      List<Map<String, Boolean>> blocks, int[][] minDistancesFromBlock
    int[] maxDistancesAtBlocks = new int[blocks.size()];
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static int apartmentHunting(List<Map<String, Boolean>> blocks
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.