Our Solution(s) Run Code

Solution 1 Solution 2

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

#include <vector>
#include <unordered_map>
#include <climits>
#include <algorithm>
#include <cmath>

using namespace std;

int getIdxAtMinValue(vector<int> array);
int distanceBetween(int a, int b);

// O(b^2*r) time | O(b) space - where b is the number of blocks and r
// number of requirements
int apartmentHunting(vector<unordered_map<string, bool>> blocks,
                     vector<string> reqs) {
  vector<int> maxDistancesAtBlocks(blocks.size(), INT_MIN);
  for (int i = 0; i < blocks.size(); i++) {
    for (string req : reqs) {
      int closestReqDistance = INT_MAX;
      for (int j = 0; j < blocks.size(); j++) {
        if (blocks[j][req]) {
          closestReqDistance = min(closestReqDistance, distanceBetween
        }
      }
      maxDistancesAtBlocks[i] =
          max(maxDistancesAtBlocks[i], closestReqDistance);
    }
  }
  return getIdxAtMinValue(maxDistancesAtBlocks);
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
#include <unordered_map>

using namespace std;

int apartmentHunting(vector<unordered_map<string, bool>> blocks,
                     vector<string> reqs) {
  // Write your code here;
  return -1;
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.