```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 is the
// 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(i, j));
        }
      }
      maxDistancesAtBlocks[i] =
          max(maxDistancesAtBlocks[i], closestReqDistance);
    }
  }
  return getIdxAtMinValue(maxDistancesAtBlocks);
}

int getIdxAtMinValue(vector<int> array) {
  int idxAtMinValue = 0;
  int minValue = INT_MAX;
  for (int i = 0; i < array.size(); i++) {
    int currentValue = array[i];
    if (currentValue < minValue) {
      minValue = currentValue;
      idxAtMinValue = i;
    }
  }
  return idxAtMinValue;
}

int distanceBetween(int a, int b) { return abs(a - b); }

```