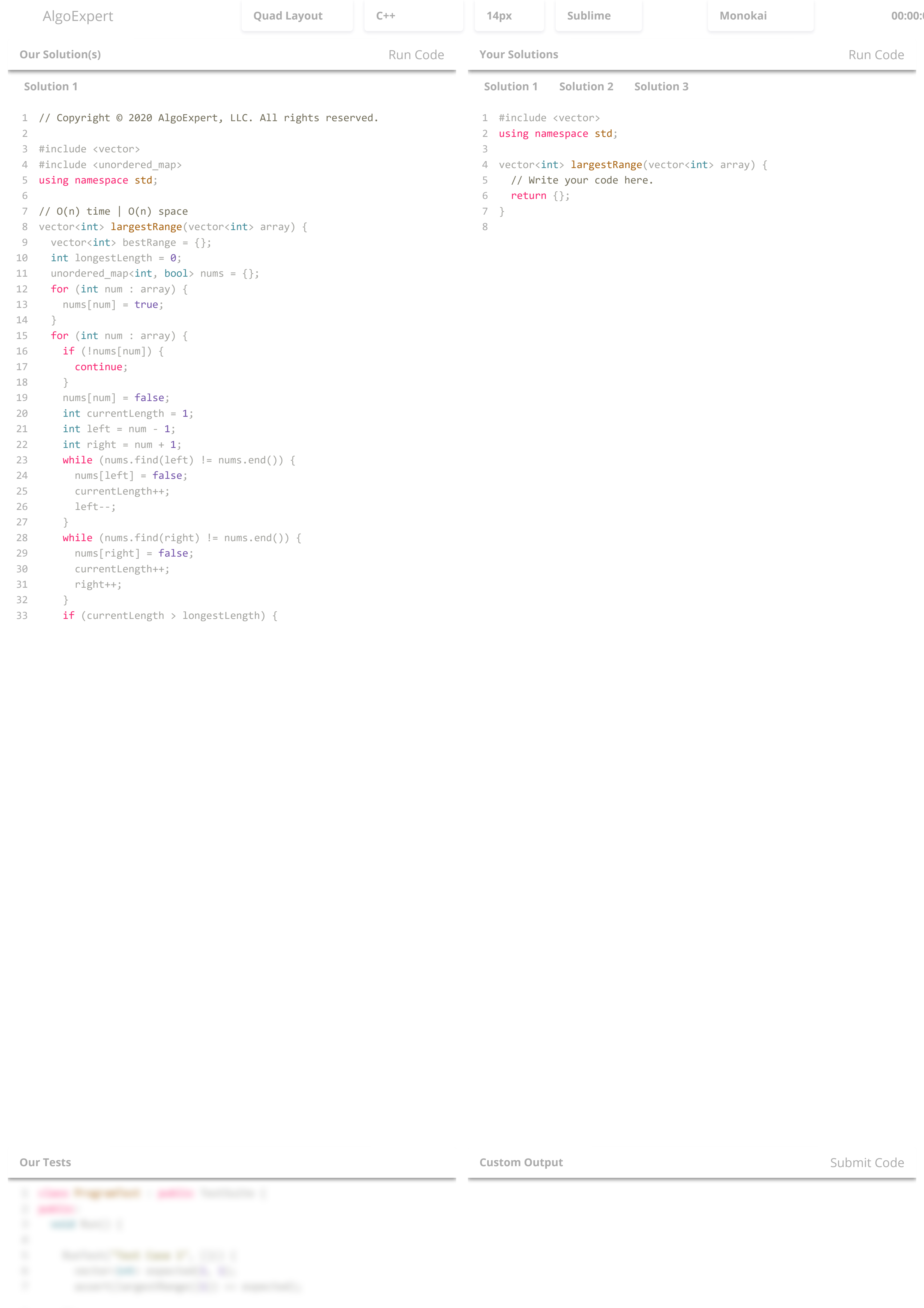

AlgoExpert

Our Solution(s)

Solution 1

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

#include <vector>
#include <unordered_map>
using namespace std;

// O(n) time | O(n) space
vector<int> largestRange(vector<int> array) {
  vector<int> bestRange = {};
  int longestLength = 0;
  unordered_map<int, bool> nums = {};
  for (int num : array) {
    nums[num] = true;
  }
  for (int num : array) {
    if (!nums[num]) {
      continue;
    }
    nums[num] = false;
    int currentLength = 1;
    int left = num - 1;
    int right = num + 1;
    while (nums.find(left) != nums.end()) {
      nums[left] = false;
      currentLength++;
      left--;
    }
    while (nums.find(right) != nums.end()) {
      nums[right] = false;
      currentLength++;
      right++;
    }
    if (currentLength > longestLength) {
```

Your Solutions

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

vector<int> largestRange(vector<int> array) {
  // Write your code here.
  return {};
}
```

Our Tests

```
[illegible]
```

Custom Output

Run or submit code when you're ready.