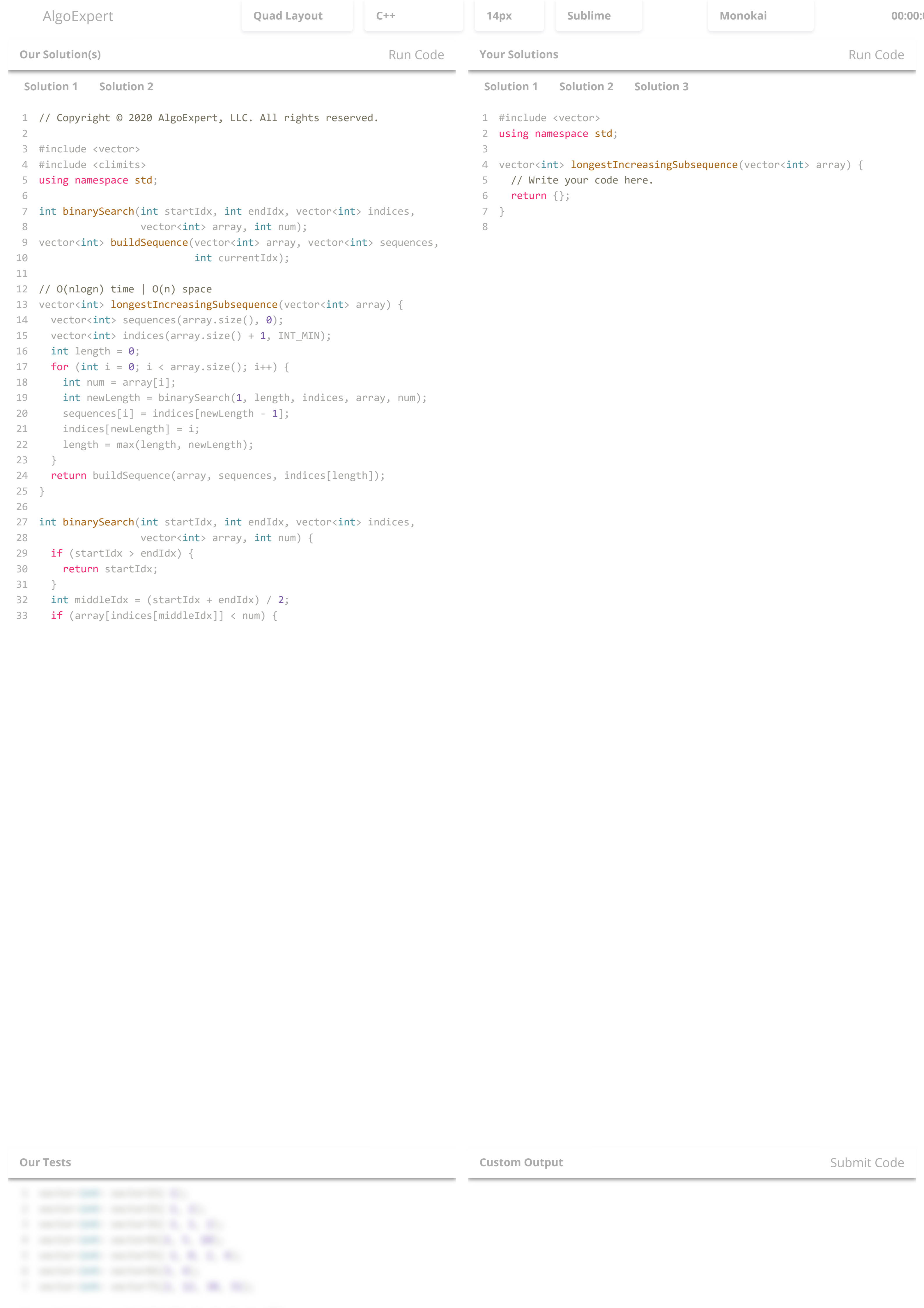

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

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

int binarySearch(int startIdx, int endIdx, vector<int> indices,
                 vector<int> array, int num);
vector<int> buildSequence(vector<int> array, vector<int> sequences,
                          int currentIdx);

// O(nlogn) time | O(n) space
vector<int> longestIncreasingSubsequence(vector<int> array) {
  vector<int> sequences(array.size(), 0);
  vector<int> indices(array.size() + 1, INT_MIN);
  int length = 0;
  for (int i = 0; i < array.size(); i++) {
    int num = array[i];
    int newLength = binarySearch(1, length, indices, array, num);
    sequences[i] = indices[newLength - 1];
    indices[newLength] = i;
    length = max(length, newLength);
  }
  return buildSequence(array, sequences, indices[length]);
}

int binarySearch(int startIdx, int endIdx, vector<int> indices,
                 vector<int> array, int num) {
  if (startIdx > endIdx) {
    return startIdx;
  }
  int middleIdx = (startIdx + endIdx) / 2;
  if (array[indices[middleIdx]] < num) {
```

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

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

```

Run or submit code when you're ready.