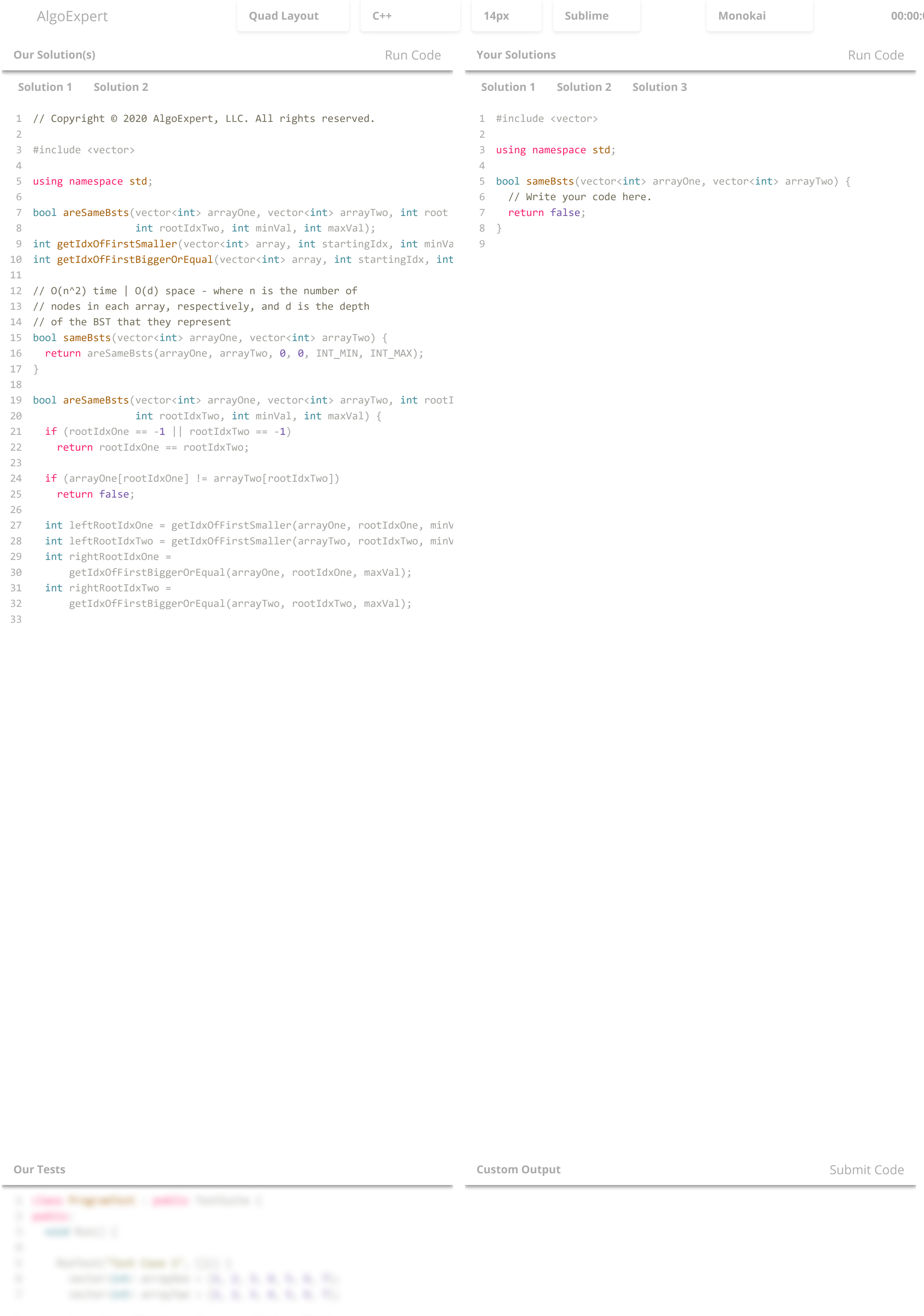

Our Solution(s) Run Code

Solution 1 Solution 2

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

#include <vector>

using namespace std;

bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int root
                 int rootIdxTwo, int minVal, int maxVal);
int getIdxOfFirstSmaller(vector<int> array, int startingIdx, int minVa
int getIdxOfFirstBiggerOrEqual(vector<int> array, int startingIdx, int

// O(n^2) time | O(d) space - where n is the number of
// nodes in each array, respectively, and d is the depth
// of the BST that they represent
bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {
  return areSameBsts(arrayOne, arrayTwo, 0, 0, INT_MIN, INT_MAX);
}

bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int rootI
                 int rootIdxTwo, int minVal, int maxVal) {
  if (rootIdxOne == -1 || rootIdxTwo == -1)
    return rootIdxOne == rootIdxTwo;

  if (arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo])
    return false;

  int leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, minV
  int leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, minV
  int rightRootIdxOne =
      getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne, maxVal);
  int rightRootIdxTwo =
      getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo, maxVal);

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>

using namespace std;

bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {
  // Write your code here.
  return false;
}

```

Our Tests

[illegible]

Custom Output Submit Code

[illegible]

Run or submit code when you're ready.