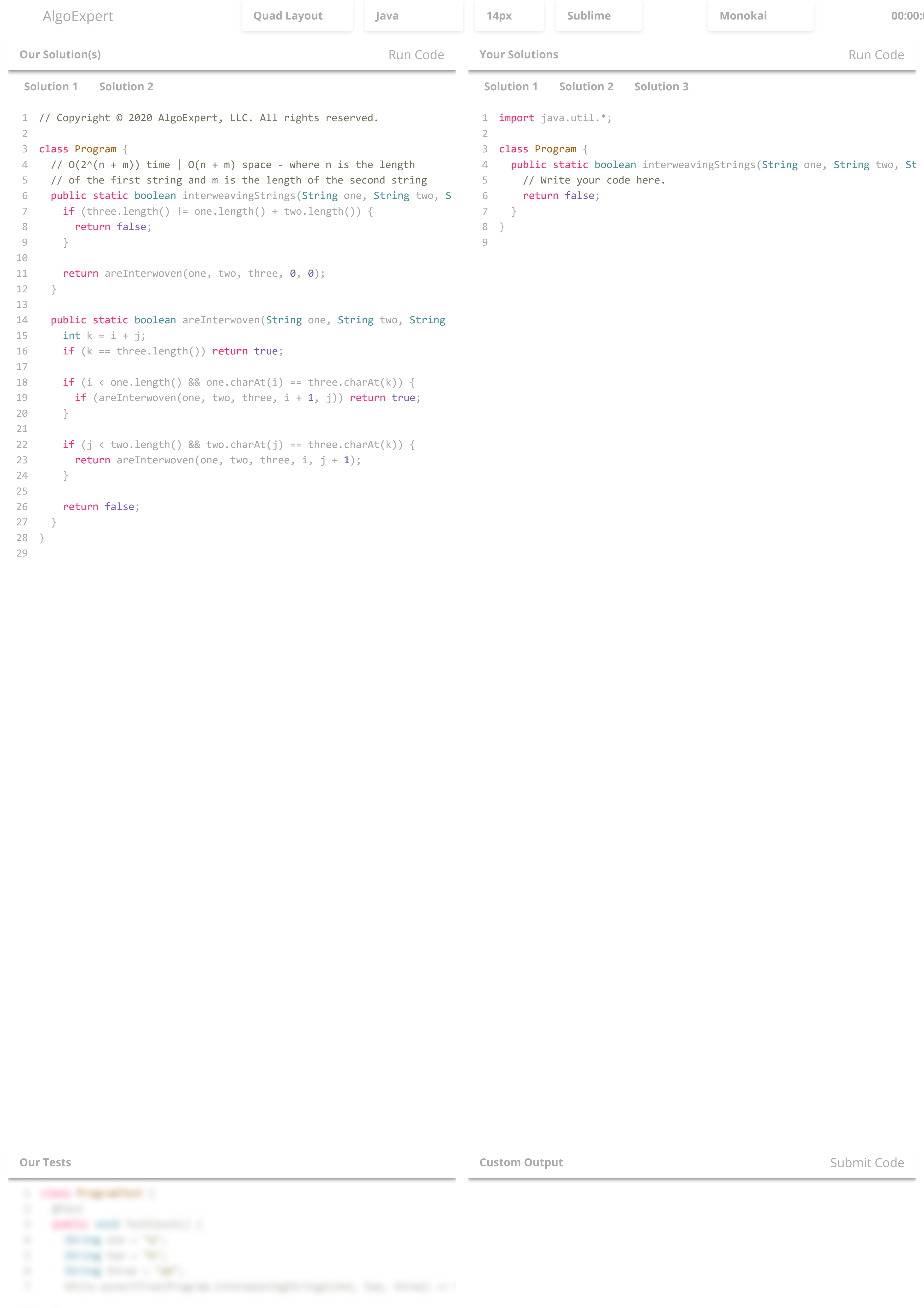

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(2^(n + m)) time | O(n + m) space - where n is the length
  // of the first string and m is the length of the second string
  public static boolean interweavingStrings(String one, String two, S
    if (three.length() != one.length() + two.length()) {
      return false;
    }

    return areInterwoven(one, two, three, 0, 0);
  }

  public static boolean areInterwoven(String one, String two, String
    int k = i + j;
    if (k == three.length()) return true;

    if (i < one.length() && one.charAt(i) == three.charAt(k)) {
      if (areInterwoven(one, two, three, i + 1, j)) return true;
    }

    if (j < two.length() && two.charAt(j) == three.charAt(k)) {
      return areInterwoven(one, two, three, i, j + 1);
    }

    return false;
  }
}

```

```java
import java.util.*;

class Program {
  public static boolean interweavingStrings(String one, String two, St
    // Write your code here.
    return false;
  }
}

```

```java
class ProgramTest {
  @Test
  public void TestCase1() {
    String one = "a";
    String two = "b";
    String three = "ab";
    Utils.assertTrue(Program.interweavingStrings(one, two, three) ==
```

Run or submit code when you're ready.