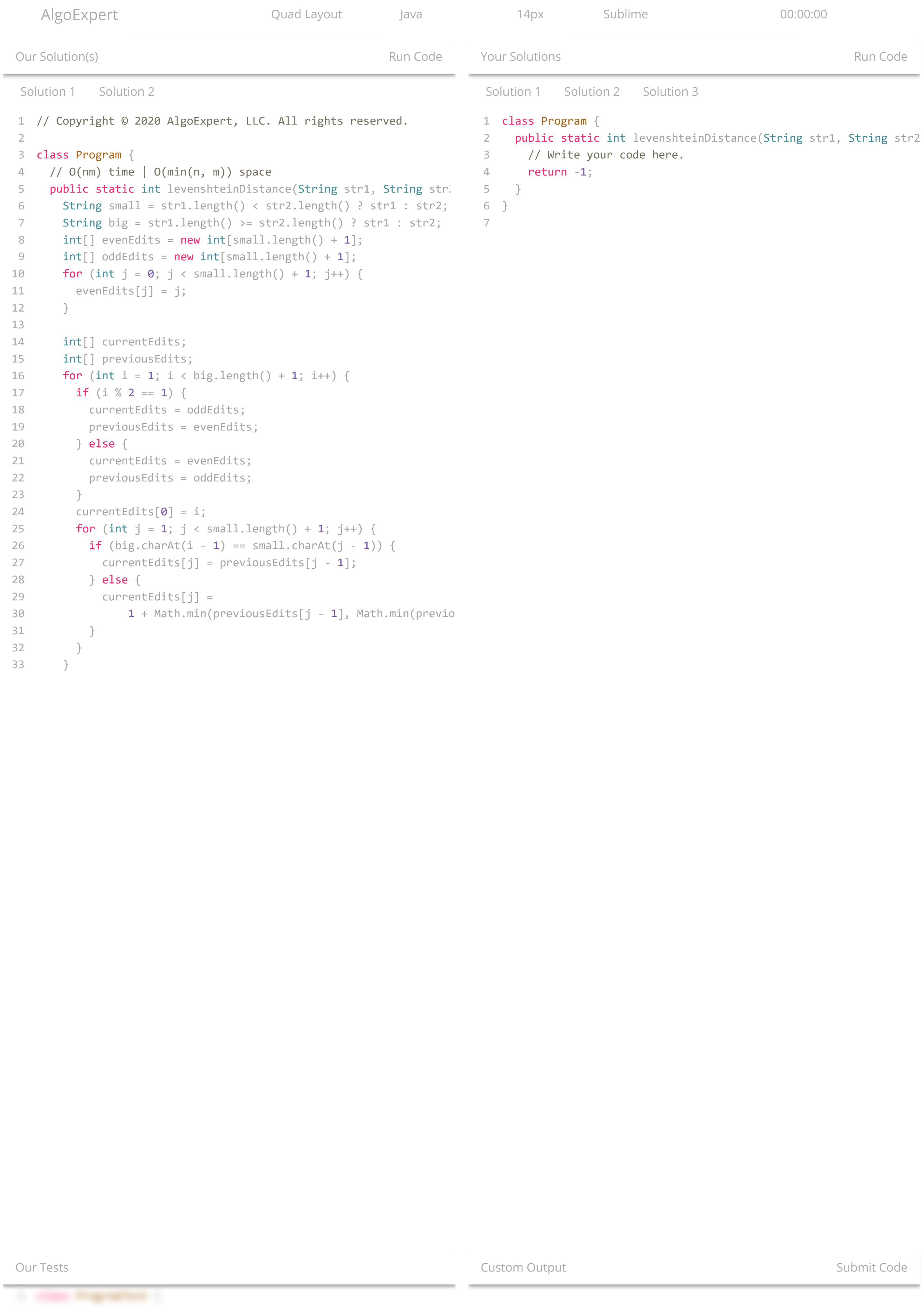

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(nm) time | O(min(n, m)) space
  public static int levenshteinDistance(String str1, String str
    String small = str1.length() < str2.length() ? str1 : str2;
    String big = str1.length() >= str2.length() ? str1 : str2;
    int[] evenEdits = new int[small.length() + 1];
    int[] oddEdits = new int[small.length() + 1];
    for (int j = 0; j < small.length() + 1; j++) {
      evenEdits[j] = j;
    }

    int[] currentEdits;
    int[] previousEdits;
    for (int i = 1; i < big.length() + 1; i++) {
      if (i % 2 == 1) {
        currentEdits = oddEdits;
        previousEdits = evenEdits;
      } else {
        currentEdits = evenEdits;
        previousEdits = oddEdits;
      }
      currentEdits[0] = i;
      for (int j = 1; j < small.length() + 1; j++) {
        if (big.charAt(i - 1) == small.charAt(j - 1)) {
          currentEdits[j] = previousEdits[j - 1];
        } else {
          currentEdits[j] =
              1 + Math.min(previousEdits[j - 1], Math.min(previo
        }
      }
    }
```

```java
class Program {
  public static int levenshteinDistance(String str1, String str2
    // Write your code here.
    return -1;
  }
}

```

Run or submit code when you're ready.