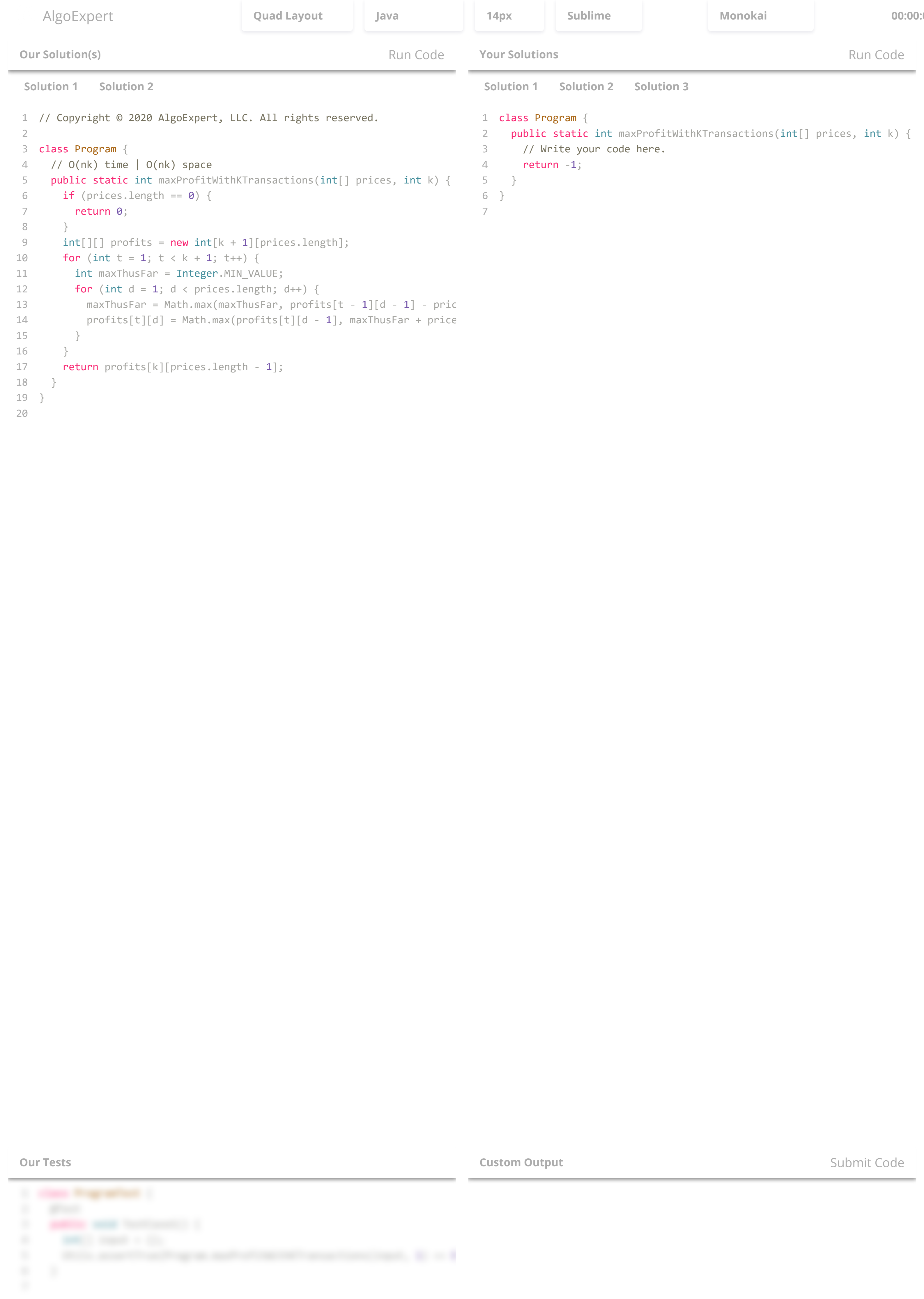

Our Solution(s)

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

class Program {
  // O(nk) time | O(nk) space
  public static int maxProfitWithKTransactions(int[] prices, int k) {
    if (prices.length == 0) {
      return 0;
    }
    int[][] profits = new int[k + 1][prices.length];
    for (int t = 1; t < k + 1; t++) {
      int maxThusFar = Integer.MIN_VALUE;
      for (int d = 1; d < prices.length; d++) {
        maxThusFar = Math.max(maxThusFar, profits[t - 1][d - 1] - pric
        profits[t][d] = Math.max(profits[t][d - 1], maxThusFar + price
      }
    }
    return profits[k][prices.length - 1];
  }
}

```

Your Solutions

```
class Program {
  public static int maxProfitWithKTransactions(int[] prices, int k) {
    // Write your code here.
    return -1;
  }
}

```

Our Tests

```
class ProgramTest {
  @Test
  public void TestCase1() {
    int[] input = {};
    Utils.assertTrue(Program.maxProfitWithKTransactions(input, 1) == [illegible]
  }

```

Custom Output

Run or submit code when you're ready.