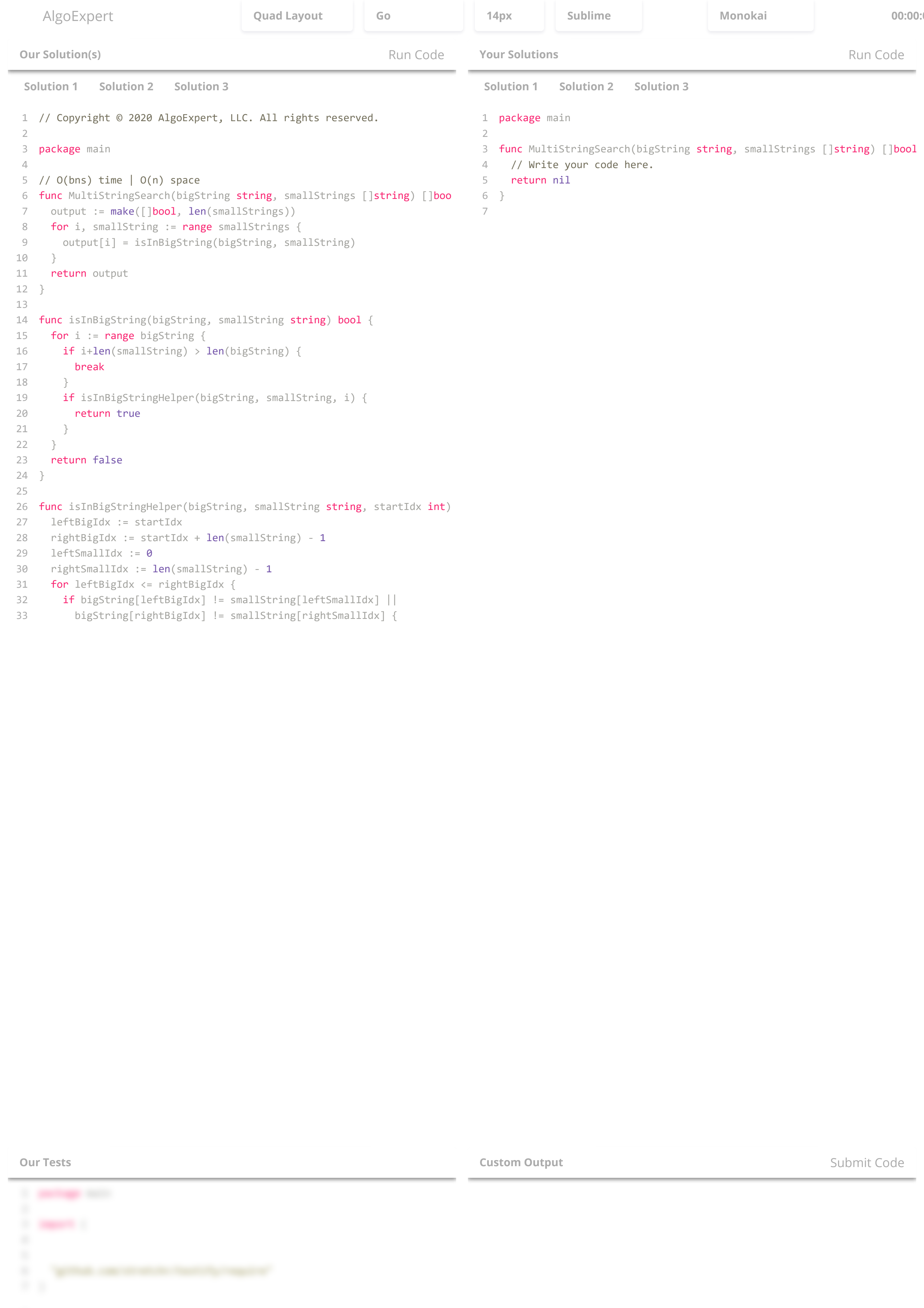

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// O(bns) time | O(n) space
func MultiStringSearch(bigString string, smallStrings []string) []boo
  output := make([]bool, len(smallStrings))
  for i, smallString := range smallStrings {
    output[i] = isInBigString(bigString, smallString)
  }
  return output
}

func isInBigString(bigString, smallString string) bool {
  for i := range bigString {
    if i+len(smallString) > len(bigString) {
      break
    }
    if isInBigStringHelper(bigString, smallString, i) {
      return true
    }
  }
  return false
}

func isInBigStringHelper(bigString, smallString string, startIdx int)
  leftBigIdx := startIdx
  rightBigIdx := startIdx + len(smallString) - 1
  leftSmallIdx := 0
  rightSmallIdx := len(smallString) - 1
  for leftBigIdx <= rightBigIdx {
    if bigString[leftBigIdx] != smallString[leftSmallIdx] ||
      bigString[rightBigIdx] != smallString[rightSmallIdx] {
```

```go
package main

func MultiStringSearch(bigString string, smallStrings []string) []bool
  // Write your code here.
  return nil
}
```

```
func (s *TestSuite) TestCase1(t *TestCase) {
  expected := []bool{true, false, true, true, false, true, false}
  output := MultiStringSearch("this is a big string", []string{"this",
  require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase2(t *TestCase) {
  expected := []bool{true, true, false, true, true, false, false}
  output := MultiStringSearch("Mary goes to the shopping center every
  require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase3(t *TestCase) {
  expected := []bool{false, false, false, false, true, false, false}
```

Run or submit code when you're ready.