Our Solution(s) Run Code

Solution 1 Solution 2

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

package main

// O(n^3) time | O(1) space
func LongestPalindromicSubstring(str string) string {
  longest := ""
  for i := range str {
    for j := i; j < len(str); j++ {
      substring := str[i : j+1]
      if len(substring) > len(longest) && isPalindrome(substrin
        longest = substring
      }
    }
  }
  return longest
}

func isPalindrome(str string) bool {
  for i := range str {
    j := len(str) - i - 1
    if str[i] != str[j] {
      return false
    }
  }
  return true
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func LongestPalindromicSubstring(str string) string {
  // Write your code here.
  return ""
}

```

Our Tests

Custom Output Submit Code

```
import (

    "github.com/stretchr/testify/require"
)

func (s *TestSuite) TestCase1() *TestCase1 {
    expected := "a"
    output := LongestPalindrome(s.Arg("a"))
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase2() *TestCase2 {
    expected := "aaaa"
    output := LongestPalindrome(s.Arg("aa"))
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase3() *TestCase3 {
    expected := "aaaa"
```

Run or submit code when you're ready.