Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3

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

package main

// O(n^2) time | O(1) space
func TwoNumberSum(array []int, target int) []int {
  for i := 0; i < len(array)-1; i++ {
    firstNum := array[i]
    for j := i + 1; j < len(array); j++ {
      secondNum := array[j]
      if firstNum+secondNum == target {
        return []int{firstNum, secondNum}
      }
    }
  }
  return []int{}
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func TwoNumberSum(array []int, target int) []int {
  // Write your code here.
  return nil
}

```

Our Tests

Custom Output Submit Code

```
}

func (s *TestSuite) TestCase15(t *testing.T) {
    expected := [][]int{0, 0}
    output := TwoDimensionSum([][]int{0, 0}, 0, 0)
    require.ElementsMatch(t, expected, output)
}

func (s *TestSuite) TestCase16(t *testing.T) {
    expected := [][]int{1, 0}
    output := TwoDimensionSum([][]int{0, 0, 1}, 1)
    require.ElementsMatch(t, expected, output)
}

func (s *TestSuite) TestCase17(t *testing.T) {
    expected := [][]int{ 0, 0}
```

**Run or submit code when you're ready.**