Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3

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

package main

import "sort"

// O(nlog(n)) time | O(1) space
func TwoNumberSum(array []int, target int) []int {
  sort.Ints(array)
  left, right := 0, len(array)-1
  for left < right {
    currentSum := array[left] + array[right]
    if currentSum == target {
      return []int{array[left], array[right]}
    } else if currentSum < target {
      left += 1
    } else {
      right -= 1
    }
  }
  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) TestCase2() *TestCase {
    expected := [][]int{0, 0}
    output := TwoSumBruteForce([]int{0, 0}, 0)
    return s.CreateTestCase(2, expected, output)
}

func (s *TestSuite) TestCase3() *TestCase {
    expected := [][]int{1, 2}
    output := TwoSumBruteForce([]int{0, 4, 3, 0}, 0)
    return s.CreateTestCase(3, expected, output)
}

func (s *TestSuite) TestCase4() *TestCase {
    expected := [][]int{2, 4}
```

Run or submit code when you're ready.