Our Solution(s) Run Code

Solution 1

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

package main

func QuickSort(array []int) []int {
  return helper(array, 0, len(array)-1)
}

func helper(array []int, start, end int) []int {
  if start >= end {
    return array
  }

  pivot := start
  left := start + 1
  right := end
  for right >= left {
    if array[left] > array[pivot] && array[right] < array[pivot] {
      array[left], array[right] = array[right], array[left]
    }
    if array[left] <= array[pivot] {
      left += 1
    }
    if array[right] >= array[pivot] {
      right -= 1
    }
  }

  array[pivot], array[right] = array[right], array[pivot]

  if right-1-start < end-(right+1) {
    helper(array, start, right-1)
    helper(array, right+1, end)
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

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

```

Our Tests

Custom Output Submit Code

```
}

func (s *TestSuite) TestCase13(t *testing.T) {
    expected := [][]int{}
    output := groupThePeople([]int{3})
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase14(t *testing.T) {
    expected := [][]int{3, 4}
    output := groupThePeople([]int{3, 4})
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase15(t *testing.T) {
    expected := [][]int{3, 4}
```

Run or submit code when you're ready.