Our Solution(s) Run Code

Solution 1

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

package main

import "math"

// O(n) time | O(1) space
func FindThreeLargestNumbers(array []int) []int {
  three := []int{math.MinInt32, math.MinInt32, math.MinInt32}
  for _, num := range array {
    updateLargest(three, num)
  }
  return three
}

func updateLargest(three []int, num int) {
  if num > three[2] {
    shiftAndUpdate(three, num, 2)
  } else if num > three[1] {
    shiftAndUpdate(three, num, 1)
  } else if num > three[0] {
    shiftAndUpdate(three, num, 0)
  }
}

func shiftAndUpdate(array []int, num int, idx int) {
  for i := 0; i < idx+1; i++ {
    if i == idx {
      array[i] = num
    } else {
      array[i] = array[i+1]
    }
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

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

```

Our Tests

Custom Output Submit Code

```
}

func (s *TestSuite) TestCase13(t *testing.T) {
    expected := []int{7, 8, 9}
    output := findPrimesLargerMedian([]int{5, 7, 8, 9})
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase14(t *testing.T) {
    expected := []int{11, 13, 15}
    output := findPrimesLargerMedian([]int{10, 11, 12, 13, 14, 15})
    require.Equal(t, expected, output)
}

func (s *TestSuite) TestCase15(t *testing.T) {
    expected := []int{11, 13, 15}
```

Run or submit code when you're ready.