Our Solution(s) Run Code

Solution 1

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

public class Program {
  // Best: O(nlog(n)) time | O(log(n)) space
  // Average: O(nlog(n)) time | O(log(n)) space
  // Worst: O(n^2) time | O(log(n)) space
  public static int[] QuickSort(int[] array) {
    QuickSort(array, 0, array.Length - 1);
    return array;
  }

  public static void QuickSort(int[] array, int startIdx, int endIdx)
    if (startIdx >= endIdx) {
      return;
    }
    int pivotIdx = startIdx;
    int leftIdx = startIdx + 1;
    int rightIdx = endIdx;
    while (rightIdx >= leftIdx) {
      if (array[leftIdx] > array[pivotIdx] && array[rightIdx] < array[
        swap(leftIdx, rightIdx, array);
      }
      if (array[leftIdx] <= array[pivotIdx]) {
        leftIdx += 1;
      }
      if (array[rightIdx] >= array[pivotIdx]) {
        rightIdx -= 1;
      }
    }
    swap(pivotIdx, rightIdx, array);
    bool leftSubarrayIsSmaller = rightIdx - 1 - startIdx < endIdx - (r
    if (leftSubarrayIsSmaller) {
      QuickSort(array, startIdx, rightIdx - 1);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
public class Program {
  public static int[] QuickSort(int[] array) {
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

```
}

[Test]
public void TestCase01() {
    int[] expected = {3, 6};
    int[] input = {3, 6};
    Write.Assert(Program.QuickSort(input), expected);
}

[Test]
public void TestCase02() {
    int[] expected = {3, 6};
    int[] input = {3, 6};
    Write.Assert(Program.QuickSort(input), expected);
}
```

Run or submit code when you're ready.