Our Solution(s) Run Code

Solution 1

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

public class Program {
  // Best: O(nlog(n)) time | O(1) space
  // Average: O(nlog(n)) time | O(1) space
  // Worst: O(nlog(n)) time | O(1) space
  public static int[] HeapSort(int[] array) {
    buildMaxHeap(array);
    for (int endIdx = array.Length - 1; endIdx > 0; endIdx--) {
      swap(0, endIdx, array);
      siftDown(0, endIdx - 1, array);
    }
    return array;
  }

  public static void buildMaxHeap(int[] array) {
    int firstParentIdx = (array.Length - 2) / 2;
    for (int currentIdx = firstParentIdx; currentIdx >= 0; currentIdx-
      siftDown(currentIdx, array.Length - 1, array);
    }
  }

  public static void siftDown(int currentIdx, int endIdx, int[] heap)
    int childOneIdx = currentIdx * 2 + 1;
    while (childOneIdx <= endIdx) {
      int childTwoIdx = currentIdx * 2 + 2 <= endIdx ? currentIdx * 2
      int idxToSwap;
      if (childTwoIdx != -1 && heap[childTwoIdx] > heap[childOneIdx])
        idxToSwap = childTwoIdx;
      } else {
        idxToSwap = childOneIdx;
      }
      if (heap[idxToSwap] > heap[currentIdx]) {
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

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

```

Our Tests

Custom Output Submit Code

```
[Test]
public void TestCase01() {
  int[] expected = {0, 0};
  int[] input = {0, 0};
  Utils.AssertTrue(comparer(Program.RangeSort(input), expected));
}

[Test]
public void TestCase02() {
  int[] expected = {0, 0};
  int[] input = {0, 0};
  Utils.AssertTrue(comparer(Program.RangeSort(input), expected));
}
```

Run or submit code when you're ready.