Our Solution(s) Run Code

Solution 1

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

using System.Collections.Generic;

public class Program {
  public class MinHeap {
    public List<int> heap = new List<int>();

    public MinHeap(List<int> array) {
      heap = buildHeap(array);
    }

    // O(n) time | O(1) space
    public List<int> buildHeap(List<int> array) {
      int firstParentIdx = (array.Count - 2) / 2;
      for (int currentIdx = firstParentIdx; currentIdx >= 0; cur
        siftDown(currentIdx, array.Count - 1, array);
      }
      return array;
    }

    // O(log(n)) time | O(1) space
    public void siftDown(int currentIdx, int endIdx, List<int> h
      int childOneIdx = currentIdx * 2 + 1;
      while (childOneIdx <= endIdx) {
        int childTwoIdx = currentIdx * 2 + 2 <=
          endIdx ? currentIdx * 2 + 2 : -1;
        int idxToSwap;
        if (childTwoIdx != -1 && heap[childTwoIdx] < heap[childO
          idxToSwap = childTwoIdx;
        } else {
          idxToSwap = childOneIdx;
        }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
using System.Collections.Generic;

// Do not edit the class below except for the buildHeap,
// siftDown, siftUp, Peek, Remove, and Insert methods.
// Feel free to add new properties and methods to the class.
public class Program {
  public class MinHeap {
    public List<int> heap = new List<int>();

    public MinHeap(List<int> array) {
      heap = buildHeap(array);
    }

    public List<int> buildHeap(List<int> array) {
      // Write your code here.
      return null;
    }

    public void siftDown(int currentIdx, int endIdx, List<int> h
      // Write your code here.
    }

    public void siftUp(int currentIdx, List<int> heap) {
      // Write your code here.
    }

    public int Peek() {
      // Write your code here.
      return -1;
    }

    public int Remove() {
      // Write your code here.
```

Our Tests

Custom Output Submit Code

```
using System.Collections.Generic;

public class Program{
    Program.MinHeap test1 = new Program.MinHeap(new int[]{
        2, 3, 5
    });

    Program.MinHeap test2 = new Program.MinHeap(new int[]{
        1, 2, 3, 4, 5, 6, 7, 8, 9
    });

    Program.MinHeap test3 = new Program.MinHeap(new int[]{
        69, 12, 24, 7, 8, -5, 24, 391, 24, 56, 2, 6, 8, 41
    });

    Program.MinHeap test4 = new Program.MinHeap(new int[]{
        -6, 5, 68, 8, -68, -6, -6, -2, -5, 3, 5, -6, -7, -1, 2, 6,
    });

    Program.MinHeap test5 = new Program.MinHeap(new int[]{
        -7, 2, 1, 6, -68, 6, -6, -68, -2, -7, 68, 5, 2, 9, -9, -1,
```

Run or submit code when you're ready.