Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class MinHeap:
    def __init__(self, array):
        self.heap = self.buildHeap(array)

    # O(n) time | O(1) space
    def buildHeap(self, array):
        firstParentIdx = (len(array) - 2) // 2
        for currentIdx in reversed(range(firstParentIdx + 1)):
            self.siftDown(currentIdx, len(array) - 1, array)
        return array

    # O(log(n)) time | O(1) space
    def siftDown(self, currentIdx, endIdx, heap):
        childOneIdx = currentIdx * 2 + 1
        while childOneIdx <= endIdx:
            childTwoIdx = currentIdx * 2 + 2 if currentIdx * 2 + 2 <= endIdx else
            if childTwoIdx != -1 and heap[childTwoIdx] < heap[childOneIdx]:
                idxToSwap = childTwoIdx
            else:
                idxToSwap = childOneIdx
            if heap[idxToSwap] < heap[currentIdx]:
                self.swap(currentIdx, idxToSwap, heap)
                currentIdx = idxToSwap
                childOneIdx = currentIdx * 2 + 1
            else:
                return

    # O(log(n)) time | O(1) space
    def siftUp(self, currentIdx, heap):
        parentIdx = (currentIdx - 1) // 2
        while currentIdx > 0 and heap[currentIdx] < heap[parentIdx]:
            self.swap(currentIdx, parentIdx, heap)
            currentIdx = parentIdx
            parentIdx = (currentIdx - 1) // 2

    # O(1) time | O(1) space
    def peek(self):
        return self.heap[0]

    # O(log(n)) time | O(1) space
    def remove(self):
        self.swap(0, len(self.heap) - 1, self.heap)
        valueToRemove = self.heap.pop()
        self.siftDown(0, len(self.heap) - 1, self.heap)
        return valueToRemove

    # O(log(n)) time | O(1) space
    def insert(self, value):
        self.heap.append(value)
        self.siftUp(len(self.heap) - 1, self.heap)

    def swap(self, i, j, heap):
        heap[i], heap[j] = heap[j], heap[i]

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
# 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.
class MinHeap:
    def __init__(self, array):
        # Do not edit the line below.
        self.heap = self.buildHeap(array)

    def buildHeap(self, array):
        # Write your code here.
        pass

    def siftDown(self):
        # Write your code here.
        pass

    def siftUp(self):
        # Write your code here.
        pass

    def peek(self):
        # Write your code here.
        pass

    def remove(self):
        # Write your code here.
        pass

    def insert(self, value):
        # Write your code here.
        pass

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.