Our Solution(s) Run Code

Solution 1

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

class MinHeap {
  constructor(array) {
    this.heap = this.buildHeap(array);
  }

  // O(n) time | O(1) space
  buildHeap(array) {
    const firstParentIdx = Math.floor((array.length - 2) / 2);
    for (let currentIdx = firstParentIdx; currentIdx >= 0; currentIdx--) {
      this.siftDown(currentIdx, array.length - 1, array);
    }
    return array;
  }

  // O(log(n)) time | O(1) space
  siftDown(currentIdx, endIdx, heap) {
    let childOneIdx = currentIdx * 2 + 1;
    while (childOneIdx <= endIdx) {
      const childTwoIdx = currentIdx * 2 + 2 <= endIdx ? currentIdx * 2 + 2 : -1;
      let idxToSwap;
      if (childTwoIdx !== -1 && heap[childTwoIdx] < heap[childOneIdx]) {
        idxToSwap = childTwoIdx;
      } else {
        idxToSwap = childOneIdx;
      }
      if (heap[idxToSwap] < heap[currentIdx]) {
        this.swap(currentIdx, idxToSwap, heap);
        currentIdx = idxToSwap;
        childOneIdx = currentIdx * 2 + 1;
      } else {
        return;
      }
    }
  }

  // O(log(n)) time | O(1) space
  siftUp(currentIdx, heap) {
    let parentIdx = Math.floor((currentIdx - 1) / 2);
    while (currentIdx > 0 && heap[currentIdx] < heap[parentIdx]) {
      this.swap(currentIdx, parentIdx, heap);
      currentIdx = parentIdx;
      parentIdx = Math.floor((currentIdx - 1) / 2);
    }
  }

  // O(1) time | O(1) space
  peek() {
    return this.heap[0];
  }

  // O(log(n)) time | O(1) space
  remove() {
    this.swap(0, this.heap.length - 1, this.heap);
    const valueToRemove = this.heap.pop();
    this.siftDown(0, this.heap.length - 1, this.heap);
    return valueToRemove;
  }

  // O(log(n)) time | O(1) space
  insert(value) {
    this.heap.push(value);
    this.siftUp(this.heap.length - 1, this.heap);
  }

  swap(i, j, heap) {
    const temp = heap[j];
    heap[j] = heap[i];
    heap[i] = temp;
  }
}

// Do not edit the line below.
exports.MinHeap = MinHeap;
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
// 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 {
  constructor(array) {
    this.heap = this.buildHeap(array);
  }

  buildHeap(array) {
    // Write your code here.
  }

  siftDown() {
    // Write your code here.
  }

  siftUp() {
    // Write your code here.
  }

  peek() {
    // Write your code here.
  }

  remove() {
    // Write your code here.
  }

  insert(value) {
    // Write your code here.
  }
}

// Do not edit the line below.
exports.MinHeap = MinHeap;
```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.