Heap Sort Algorithm:

```
class MaxBinaryHeap {
    constructor(){
        //this.values = [41,39,33,18,27,12]
        this.values = []


    }
    insert(element){
        this.values.push(element)
        this.bubbleUp()
    }
    bubbleUp(){
        let idx = this.values.length - 1
        const element = this.values[idx]

        while(idx>0){
            let parentIdx = Math.floor((idx-1)/2)
            if(this.values[parentIdx]<element){
                [this.values[parentIdx], this.values[idx]] =[this.values[idx],
this.values[parentIdx]]
                idx = parentIdx
            }else{
                break
            }

        }
    }
    bubbleDown(){
        let idx = 0
        const length = this.values.length
        const element = this.values[0]
        while(true){
            let leftChildIdx = 2 * idx + 1
            let rightChildIdx = 2 * idx + 2
            let leftChild, rightChild
            let swap = null

            if(leftChildIdx < length){
                leftChild = this.values[leftChildIdx]
                if(leftChild > element){
                    swap = leftChildIdx
                }
            }
            if(rightChildIdx < length){
                rightChild = this.values[rightChildIdx]
                if(
                    (swap === null && rightChild > element) ||
                    (swap !== null && rightChild > leftChild)
                ){
                    swap = rightChildIdx
                }
            }
```

```
            if(swap === null) break
            this.values[idx] = this.values[swap]
            this.values[swap] = element
            idx = swap

        }
    }
    sort(){
        let element = []
      while(this.values.length>0)  {
        [this.values[0], this.values[this.values.length - 1]] =
[this.values[this.values.length - 1], this.values[0]]
        element.push(this.values.pop())
        this.bubbleDown()
        }
       return element
    }
}


let heap = new MaxBinaryHeap()
heap.insert(41)
heap.insert(155)
heap.insert(15)
heap.insert(18)
heap.insert(39)
heap.insert(12)
heap.insert(55)
console.log(heap.values)
console.log(heap.sort())
```