Building a Min Binary Heap:

```
class MinBinaryHeap {
  //Add It To End
  //Bubble It Up

  constructor() {
    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);
      let parent = this.values[parentIdx];

      if (parent <= element) break;

      [this.values[parentIdx], this.values[idx]] = [
        this.values[idx],
        this.values[parentIdx],
      ];

      idx = parentIdx;
    }
  }
}

let heap = new MinBinaryHeap();
heap.insert(41)
heap.insert(155)
heap.insert(15)
heap.insert(18)
heap.insert(39)
heap.insert(12)
heap.insert(55)
heap.insert(10)
console.log(heap.values);

//[41,39,33,18,27,12,55]
```

Finding the Longest Consecutive Sequence in an Array:

```
let arr = [2,5,7,8,9,10,12,18]

function lcs(arr){
    let maxCount = 0;
    let startIndex = 0, endIndex = 0;
    for(let i = 0; i<arr.length; i++){
      let count = 1
      for(let j=i; j<arr.length; j++){
        if(arr[j+1]-1 === arr[j]){

            count++
            if(count > maxCount){
                maxCount = count
                startIndex = i
                endIndex = j
            }
        }else{
            i=j
            count == 0
        }
      }
    }
    return {max: maxCount, element: arr.slice(startIndex, endIndex+1)}
}

console.log(lcs(arr))
```