

2a .

Algorithm InsertItem (7)

Heap.push(7)

If |heap| >1 then

Upheap()

Return heap;

Algorithm Upheap()

i🡨heap.length-1

Element🡨heap[i]

parentIndex🡨Math.floor((i-1)/2)

while (i!==0 && element < heap[parentIndex]) do

swap(heap, i, parentIndex)

i🡨parentIndex

parentIndex🡨Math.floor((i-1)/2)

2b.

Algorithm removeMin()

If |heap | ==0 then return undefined

If |heap| ===1 then

Let removed 🡨heap[0]

Heap 🡨 [ ]

Return removed

swap(heap, 0, |heap|-1)

removed🡨heap.pop()

downheap();

return removed

algorithm downheap()

i🡨0

element🡨heap[0]

minIndex🡨min(heap[i\*2+1], heap [ i\*2+2])

while(minIndex <|heap| && element> heap[minIndex}

swap(heap, I , minIndex)

i=minIndex

minIndex 🡨min(heap[i\*2+1], heap [ i\*2+2])

3.

reportLower(*x*) {

*let* newArr = [];

    for (*let* i = 0; i < *this*.values.length; i++) {

*let* element = *this*.valies[i];

      if (element <= x) {

        newArr.push(element);

      } else break;

    }

    return newArr;

  }