Heaps (Priority Q)

$A = [7, 12, 9, 25, 30, 15, 40, 50, 60]$

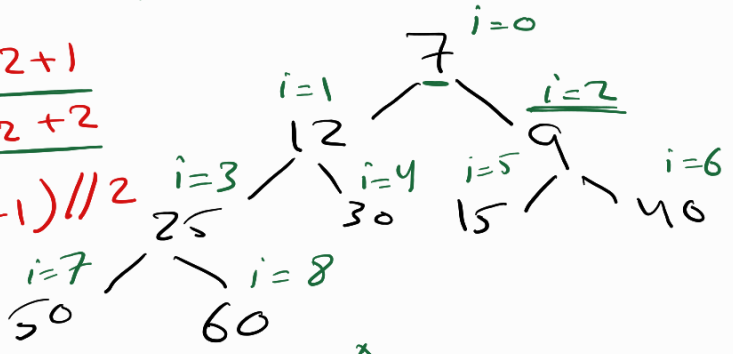
(1 2 3 4 5 6 7 8)

i=0

left_index = i*2+1

right_index = i*2+2

root_index = (i-1)//2



root

i=6

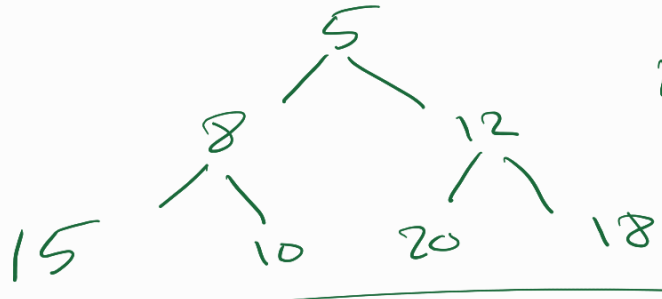
(6-1)//2

5//2

= 2

is this a min-heap

$A = [5, 8, 12, 15, 10, 20, 18]$



min-heap

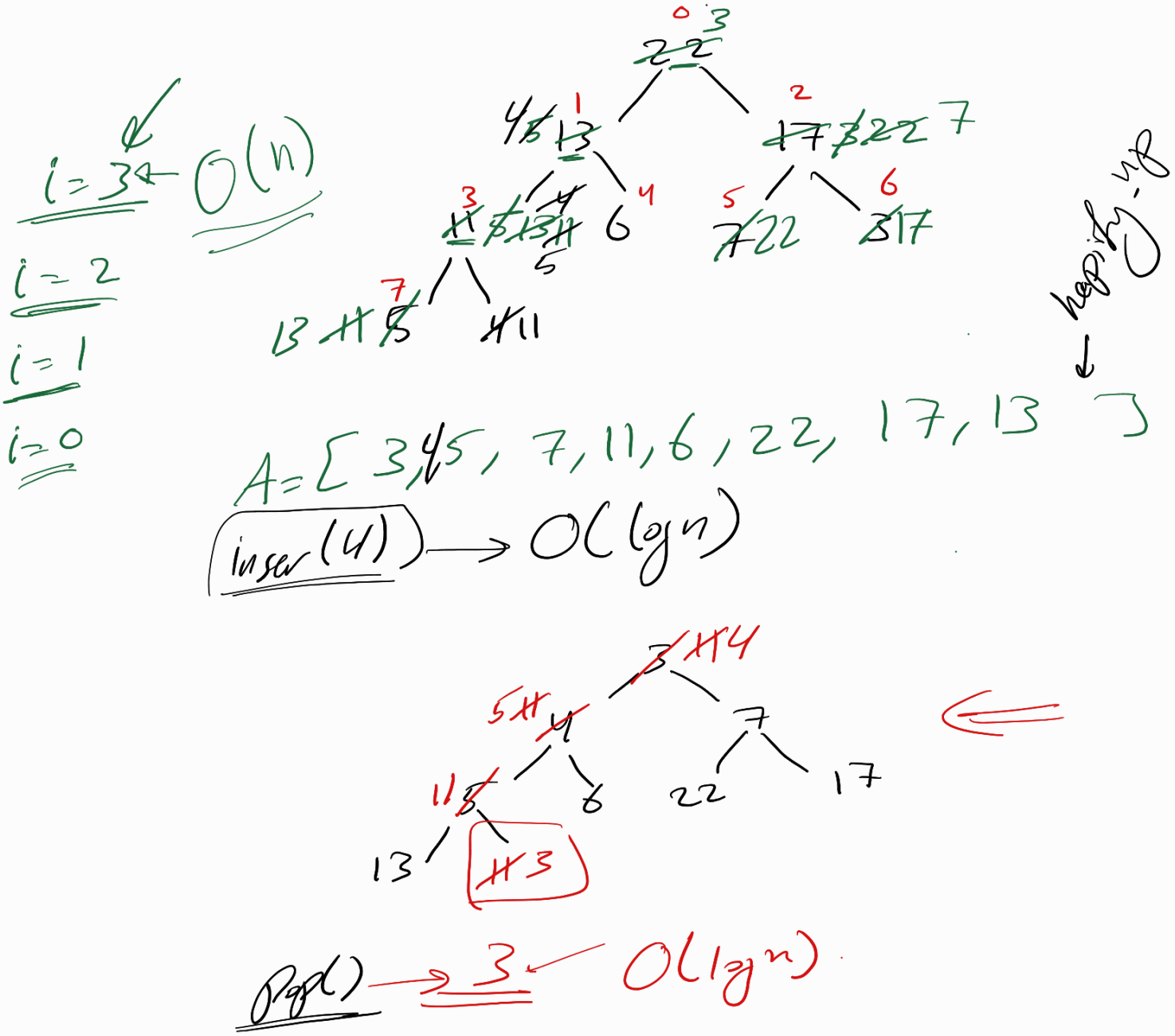
$A = [22, 13, 17, 11, 6, 7, 3, 5]$ n=8

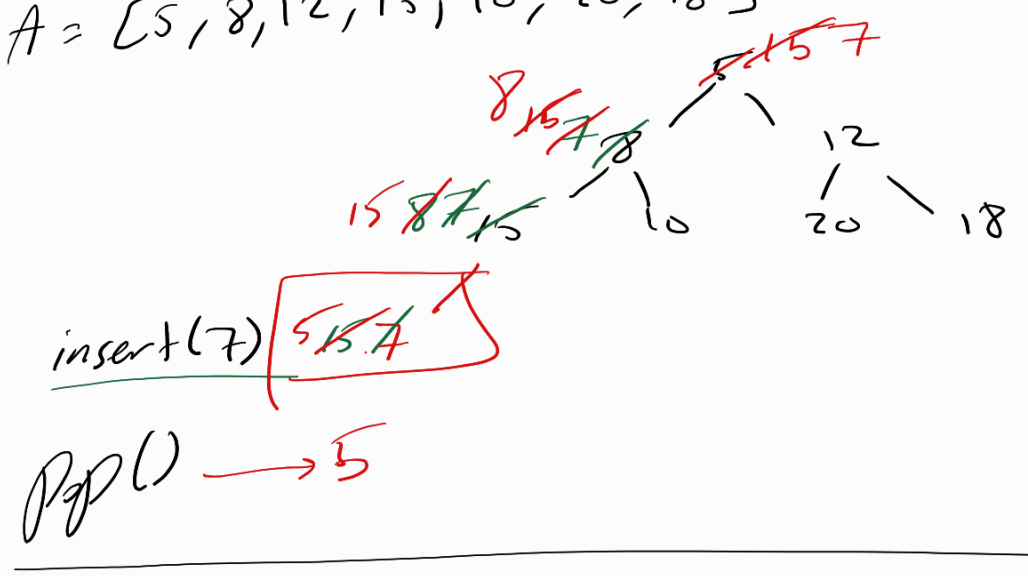
Build heap | Last internal index = $\lfloor (n-2)/2 \rfloor$

Last internal index = $\lfloor (8-2)/2 \rfloor$

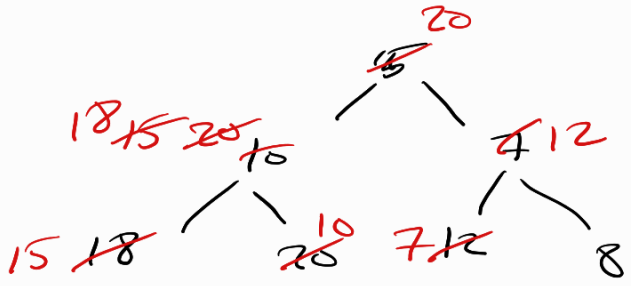
= 3

↳ heapify_down(3 → 0)


i = 3
O(n)
i = 2
i = 1
i = 0
heapify-up
A = [3, 5, 7, 11, 6, 22, 17, 13]
insert(4) → O(log n)
pop() → 3 ← O(log n)


A = [5, 8, 12, 15, 10, 20, 18]
insert(7)
pop() → 5

$A = [15, 10, 7, 18, 20, 12, 8]$

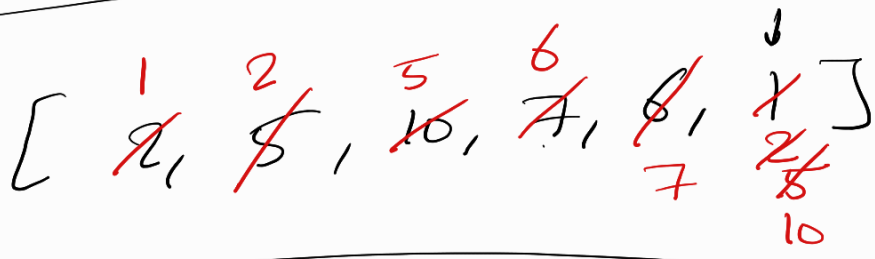


Last internal idx = floor((len(A) − 2)/2)

$= \left\lfloor \frac{7-2}{2} \right\rfloor = 2, 1, 0$

$A = [20, 18, 12, 15, 10, 7, 8]$ max-heap

Selection Sort



$[1, 2, 5, 6, 7, 10]$

$O(n^2)$

```
def selection_sort(a):
    for i in range(len(a)-1):
        m = i
        for j in range(i+1, len(a)):
```

```
    if a[j] < a[m]:
        m = j
    a[i], a[m] = a[m], a[i]
return a
```

We can enhance selection sort using heap.

① Build min-heap $\Longrightarrow O(n)$

② Repeatedly pop from heap & push to array $\Longrightarrow O(\log n * n) \Longrightarrow O(n \log n)$

$O(n) + O(n \log n) = O(n \log n)$