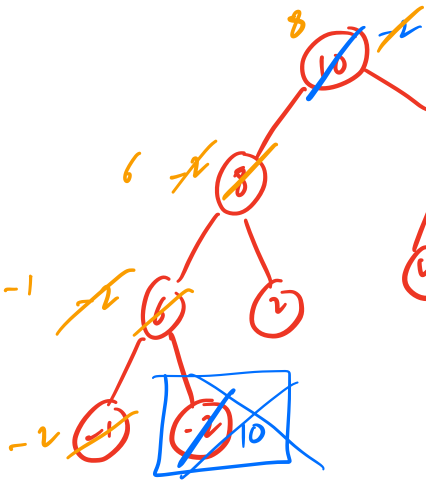
Q Given a max-heap. Can you delete the MAX element?



A: [~~10~~, ~~8~~, 5, ~~6~~, 2, 4, 1, ~~-1~~, ~~-2~~] ~~10~~

~~-2~~ ~~-2~~ ~~-2~~ -2

8 6 -1

Steps:

1) Swap(A[0], A[A.size()-1]) → O(1)

2) A.pop_back(); → O(1)

3) heapify(0); → O(log N)

$\boxed{TC = O(\log N)}$

Heap Sort

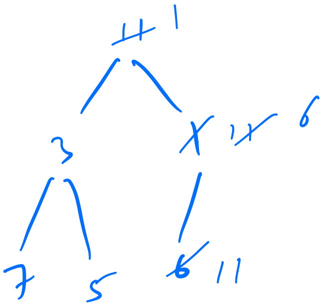
TC = O(N log N)

SC = O(1)

A : [11, 3, 1, 7, 5, 6]

MIN HEAP : O(N)

A : [1, 3, 6, 7, 5, 11]

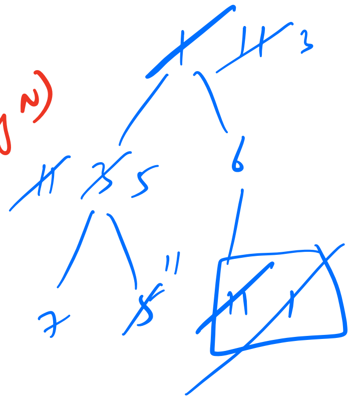


ANS : [1]

getMin() → O(1)

1

delMin() → O(log N)



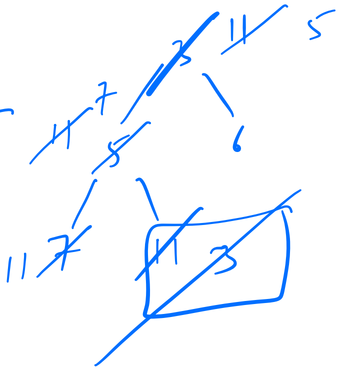
A : [3, 5, 6, 7, 11]

getMin()

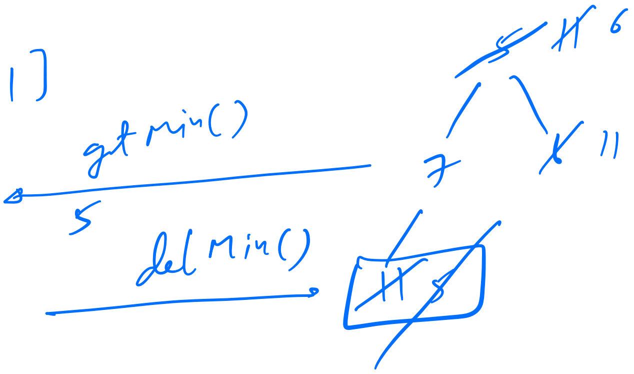
ANS : [1, 3]

3

delMin()

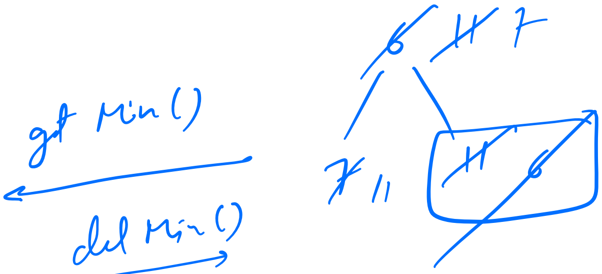


A : [5, 7, 6, 11]



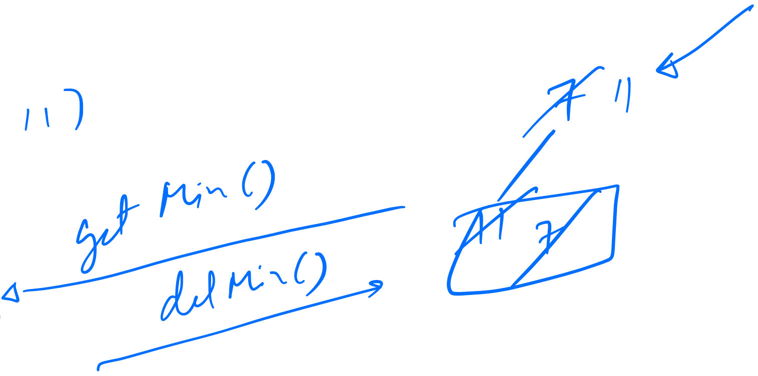
ANS : [1, 3, 5]

A : [6, 7, 11]



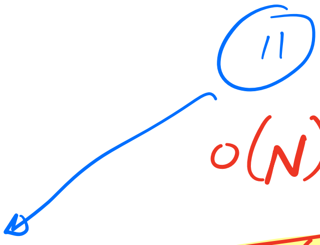
ANS: [1, 3, 5, 6]

A : [7, 11]



ANS: [1, 3, 5, 6, 7]

A: [11]



ANS: [1, 3, 5, 6, 7, 11]

$O(N) + N \times \log N$

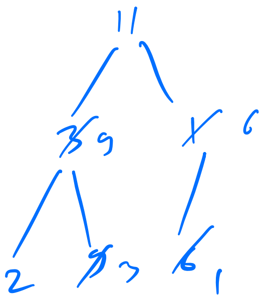
$\boxed{TC = O(N\log N)}$

$\boxed{SC = O(N)}$

II Create a MAX heap.

A: [11, 3, 1, 2, 9, 6]

↓ MAX heap:



A: [11, 9, 6, 2, 3, 1]

N: A.size()

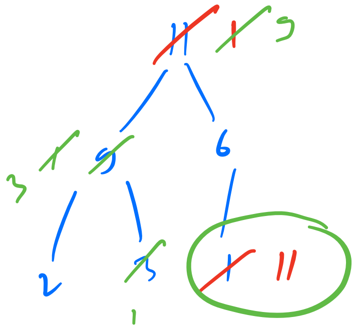
del MAX() → N--

A: [~~11~~, ~~9~~, 6, 2, ~~3~~, | ~~1~~]

~~1~~ 3 1 11

9

MAX he

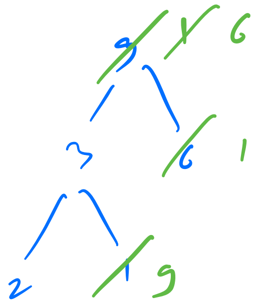


del MAX()

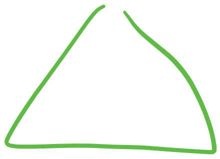
A: [~~9~~, 3, ~~6~~, 2 | ~~1~~, 11]

~~1~~ 1 9

6



A: [6, 3, 1, 2 | 9, 11]



STEPS:

// A[]

1) N = A.size();

2) Convert A[] → MAX heap. → $O(N)$

3) swap(A[0], A[N-1]); → $O(1)$

4) N--; → $O(1)$

5) heapify(0) → $O(\log N)$

$\times N$

N>0

del MAX() → N

heapify() → N

$TC = N + N \times \log N$

$\boxed{TC = O(N \log N)}$

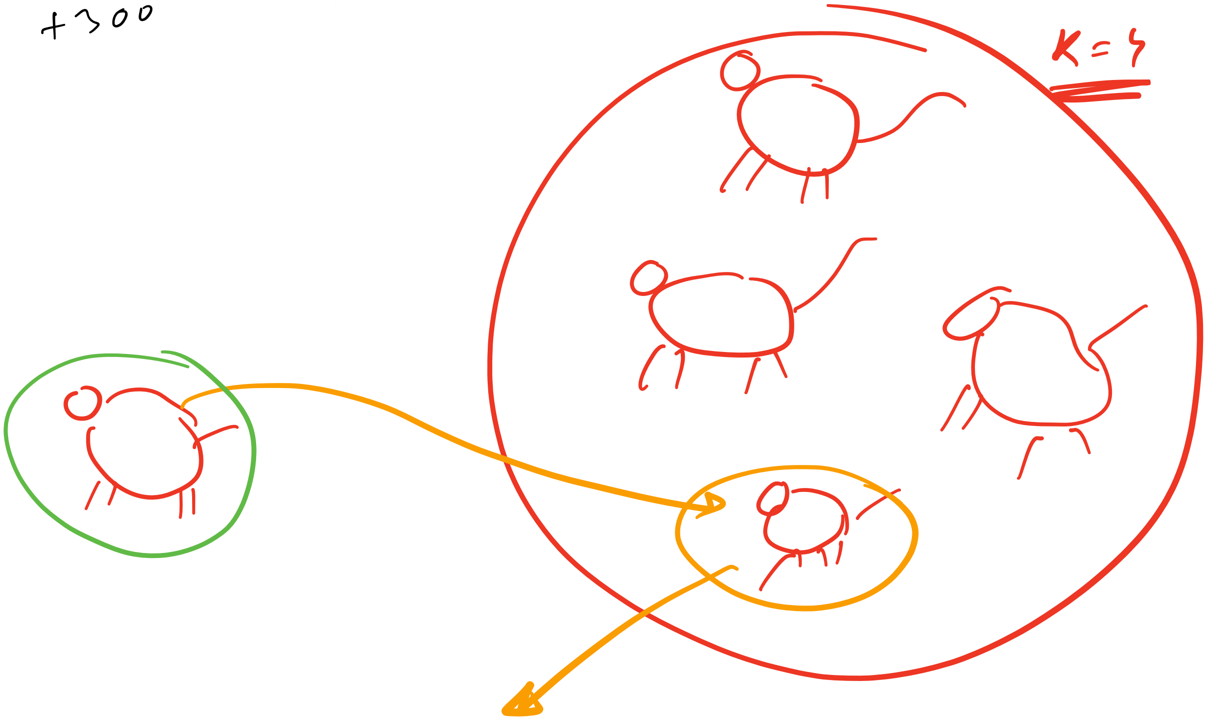
$\boxed{SC = O(1)}$

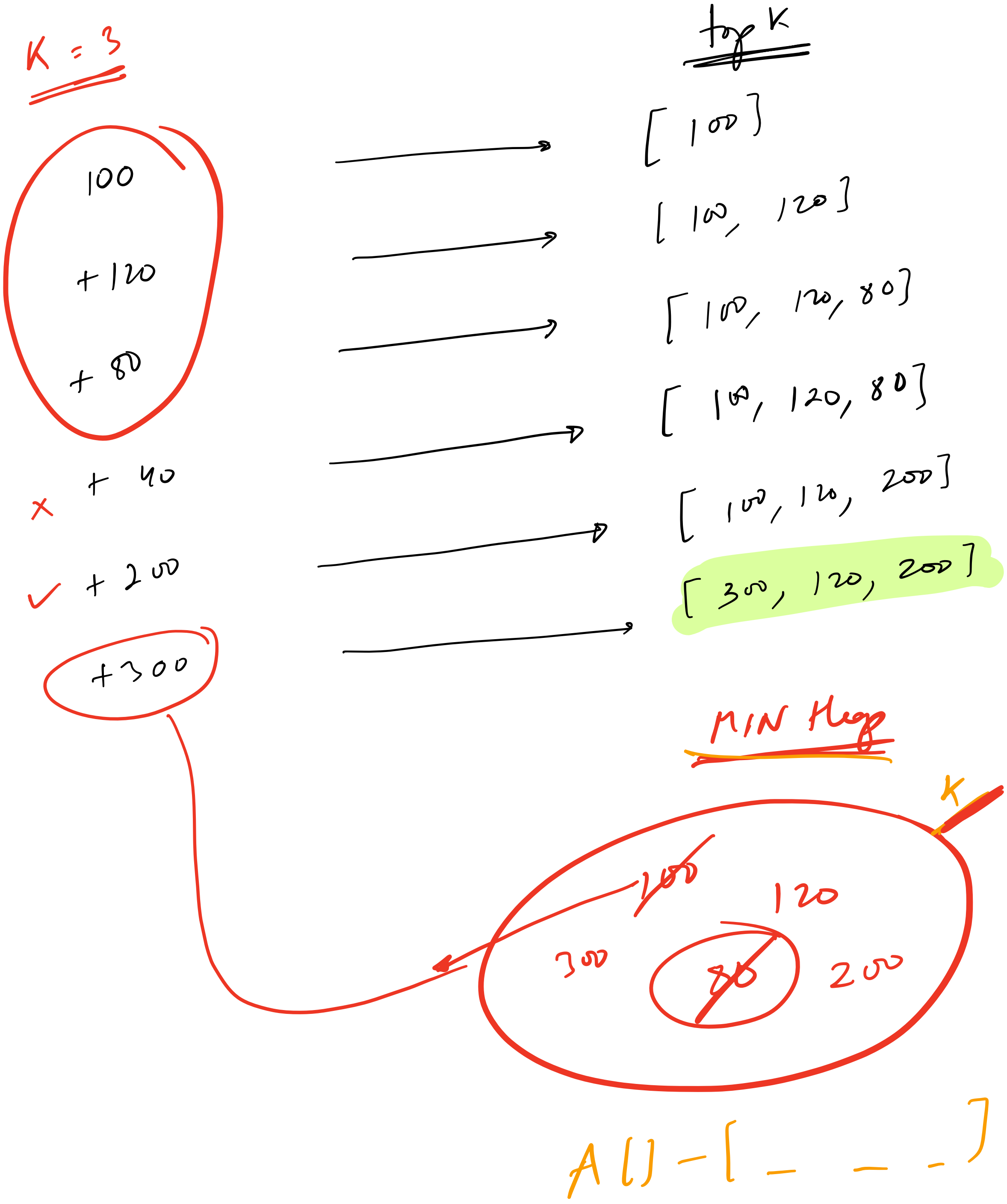
~~Q~~ Given a stream of integers.
At any point, give the largest K elements till that point!

fixed!

K = 3

	top K
100 →	[100]
+120 →	[100, 120]
+80 →	[100, 120, 80]
+ 40 →	[100, 120, 80]
+ 200 →	[100, 120, 200]
+300 →	[300, 120, 200]




K = 3
top K
100
+120
+ 80
[100]
[100, 120]
[100, 120, 80]
[100, 120, 80]
× + 40
[100, 120, 200]
✓ + 200
[300, 120, 200]
+300
MIN Heap
K
~~100~~
120
300
~~80~~
200
A[] - [_ _ _]

STEPS

MAINTAIN A MIN HEAP

+ x

```
if(MH.size() < K) {
        MH.insert(x);
}
else {
    if( MH.getMin() < x)
        MH.delMin();
        MH.insert(x);
    }
}
```

log(K)

getTopK()

```
ret MH[];
```

NOTE : MH.size() would always <= K

~~SC : O(K)~~

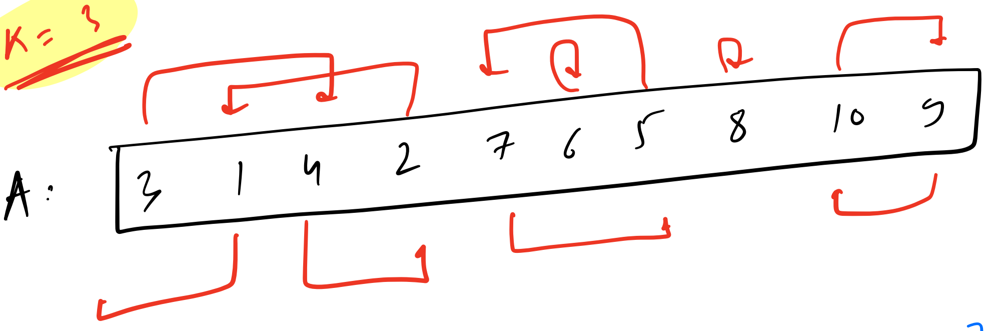
O(1)

Q Given an almost sorted array.

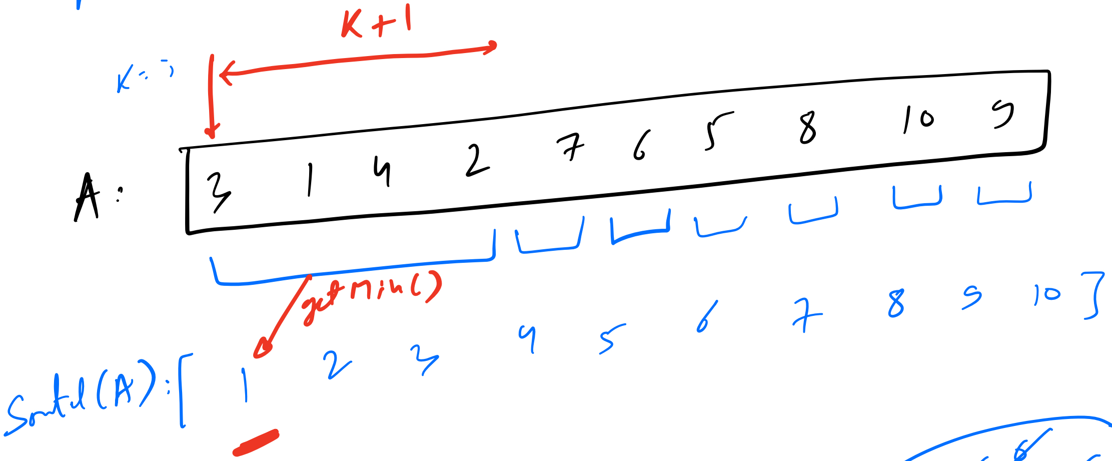
Sort it!

↳ Every element is at MAX K distance from it's sorted pos!

K = 3



A : [1 2 3 4 5 6 7 8 9 10]



Create a MIN H {size K+1

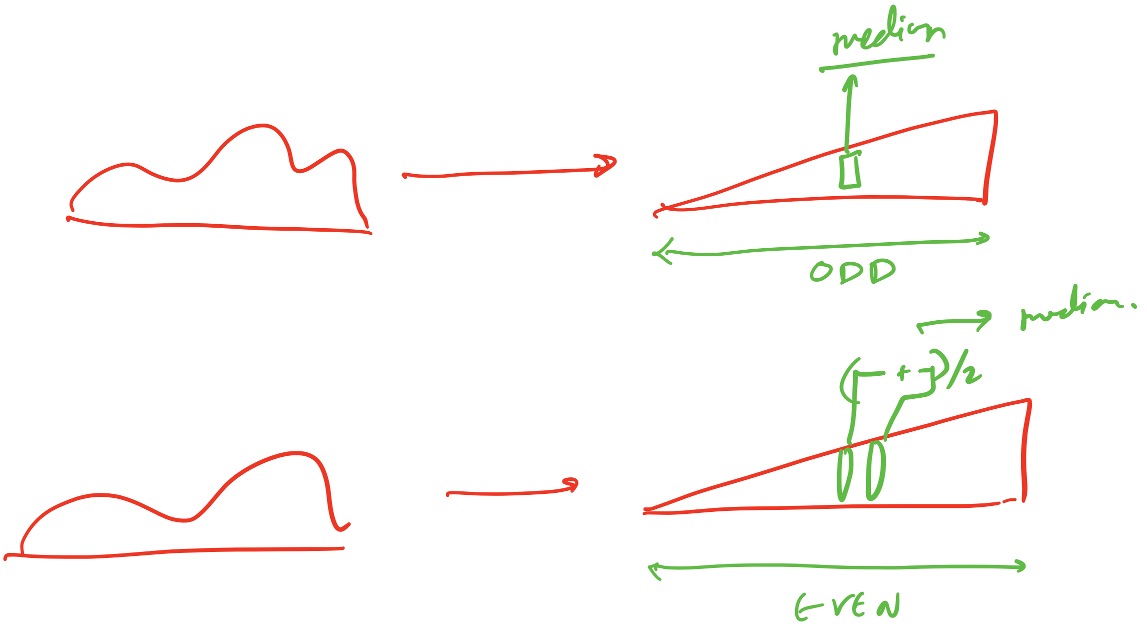
↳ O(K)

N-1 × log(K)

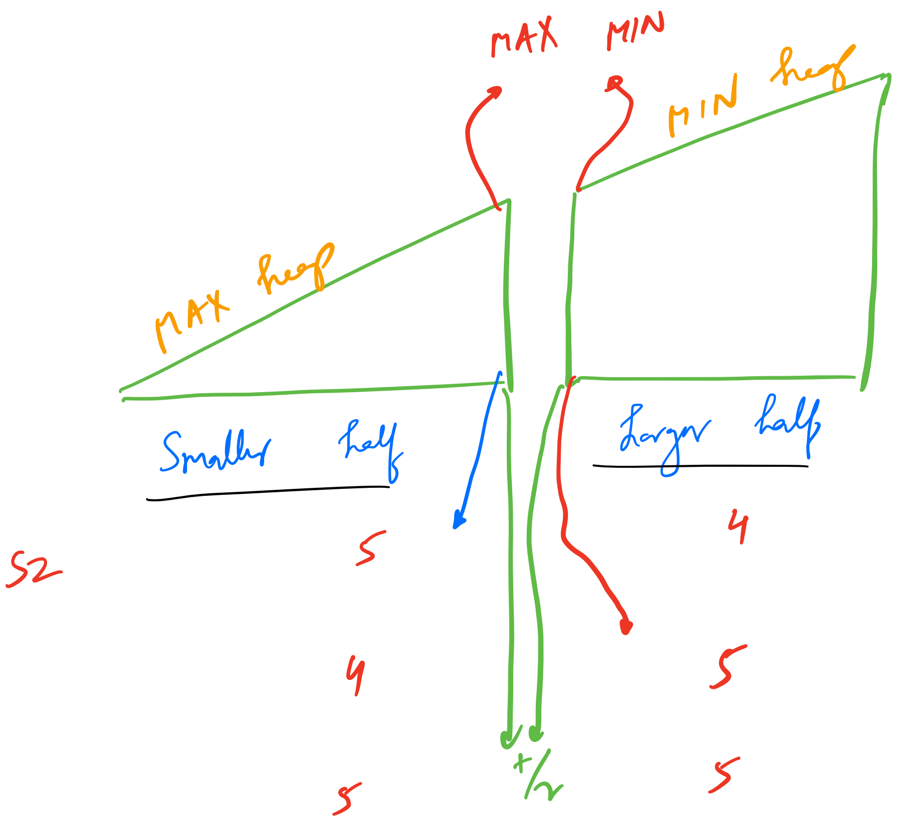
TC: O(N log K)

SC: O(K)

Q Given a stream of integers.
Find the median at every point!



		Median
+10	[10]	10
+20	[10, 20]	15
+1	[1, 10, 20]	10
+7	[1, 7, 10, 20]	8.5
+11	[1, 7, 10, 11, 20]	10

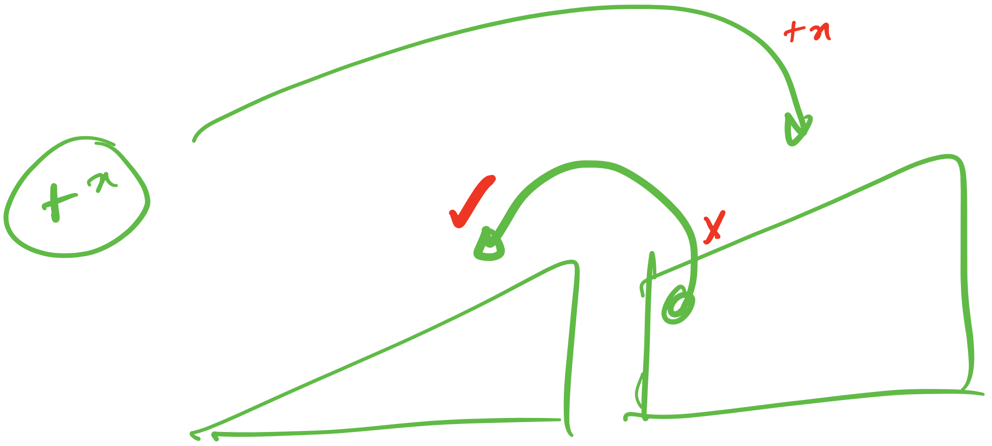


	MAX	MIN	MEDIAN
+10	10		10
+20	10	20	15
+1	1 10	20	10

+2 [1 2] | [10 20] 6

+30 [1 2] | [10 20 30] 10

+15 [1 2 10] [15 20 30] 12.5



$TC = O(\lg N)$

$SC = O(N)$