Q. Given an array, find the K smaller elements. (print)

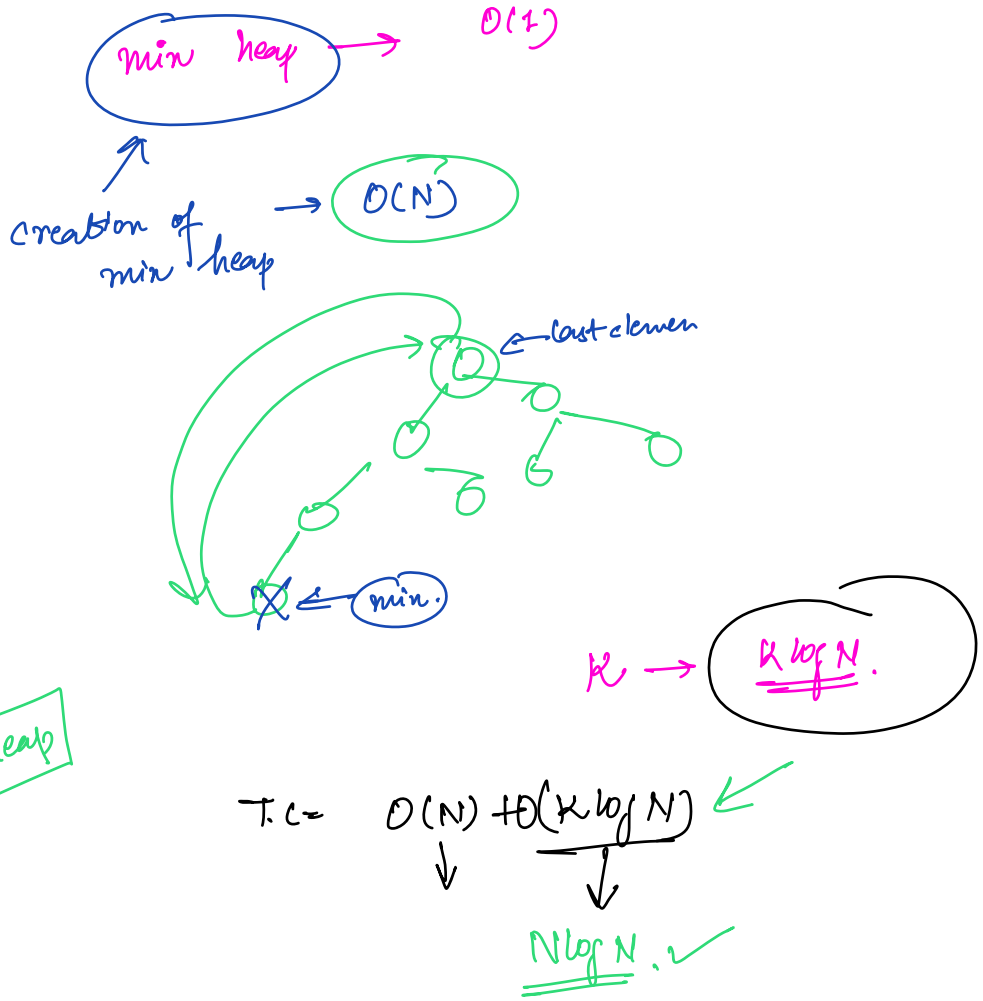
I/P :- A : [8, 3, 10, 4, 11, 2, 7, 6, 5, 1], K = 4

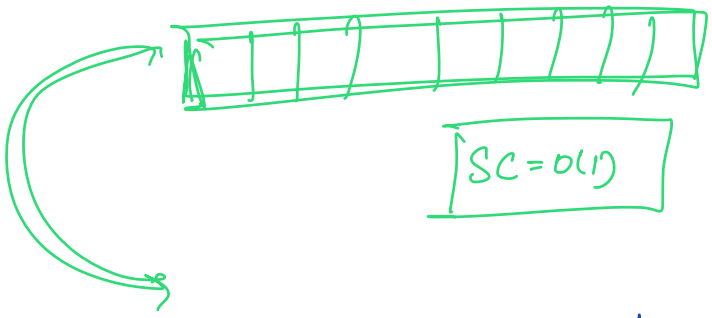
O/P :- 1, 2, 3, 4

Approach 1.

T.C = O(nlogn)

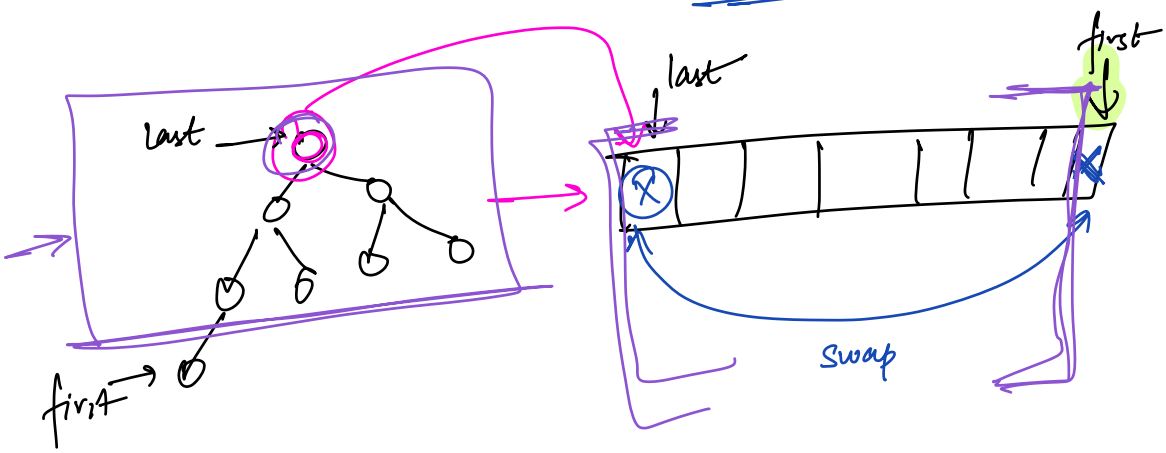
O(N) / O(1) O(logn)

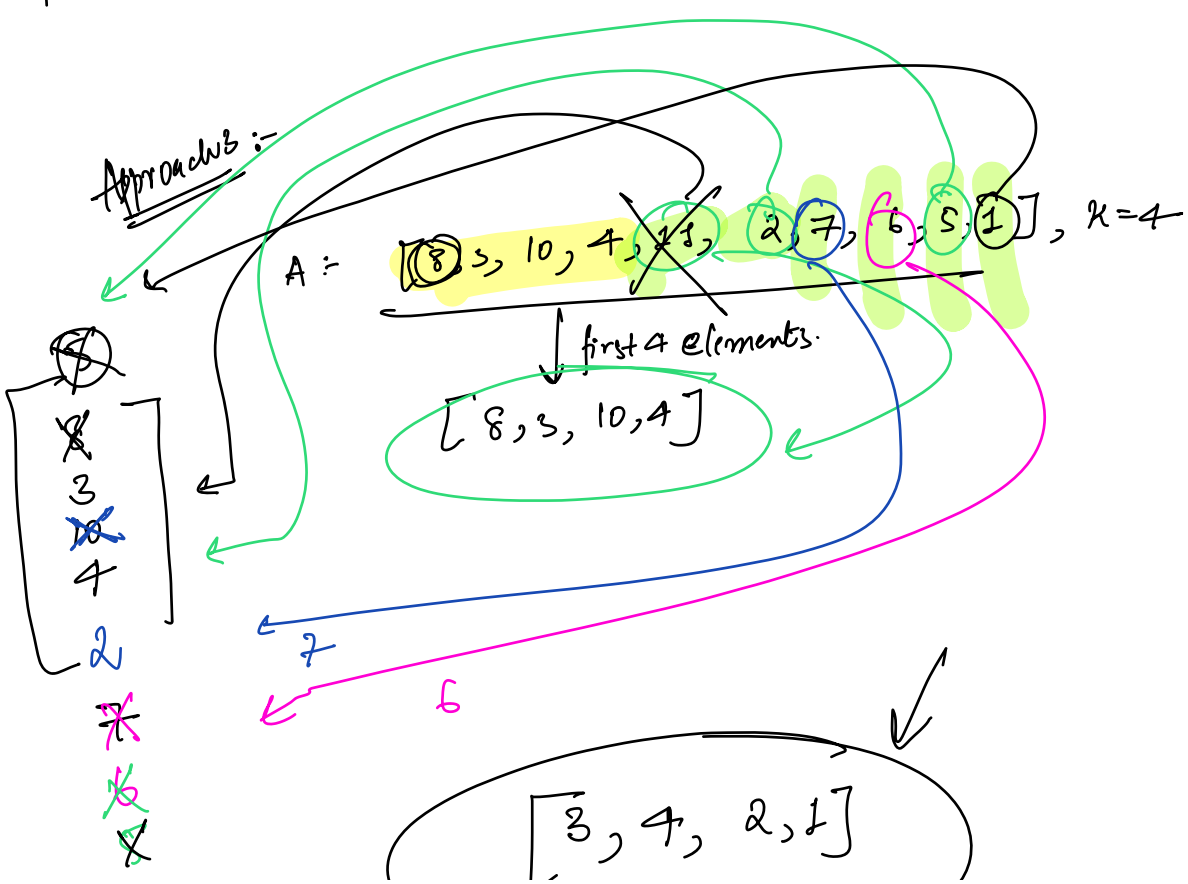




* Cannot modify the array.

SC = O(N).

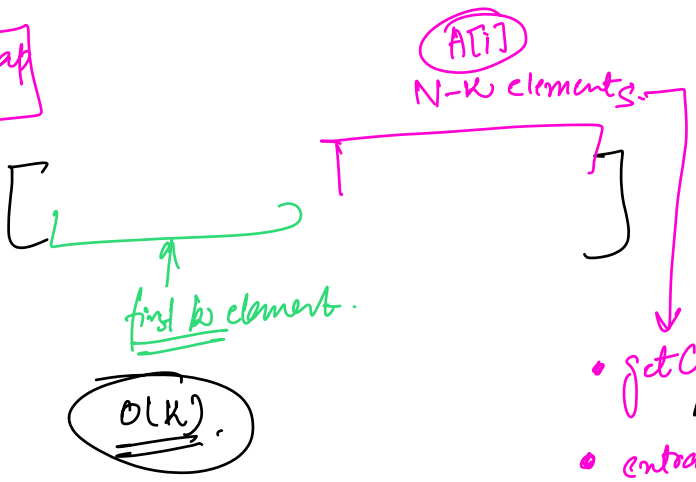


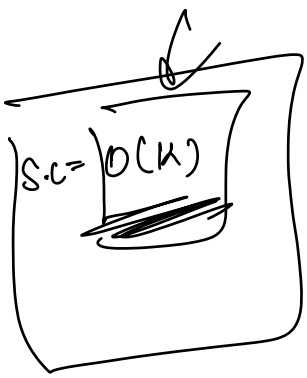


1.

A[i] < max of current group

If yes. remove (max of current group)
insert A[i] into the group.





- Get Current Max
  ↳ O(1)
- extract Max
  ↳ log K.
- insert A[i]
  ↳ log K

(N-K) * (2 log K)

T.C = O(K) + (N-K) * log K

N log N

Q K sorted Array.

Given an array. It is nearly sorted.

↳ . K nearly sorted.

↓

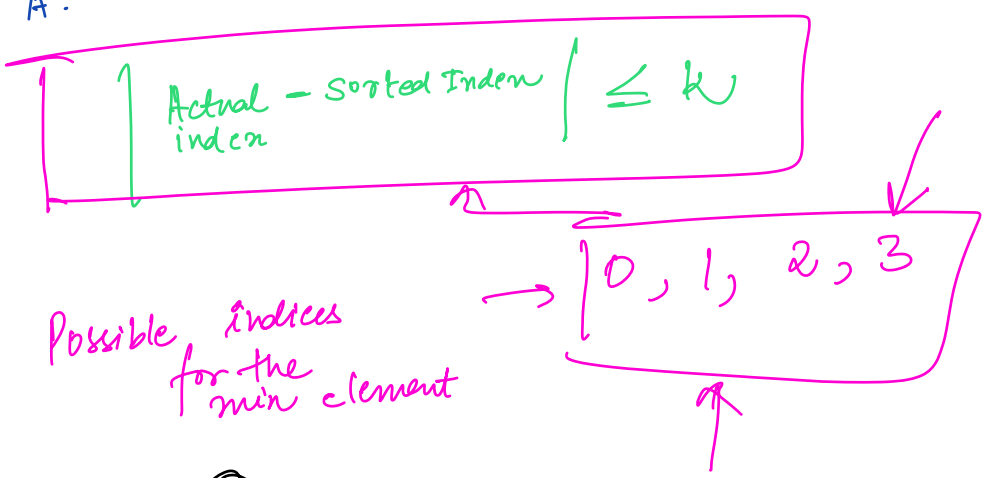
every element is atmost k positions away from its sorted position.

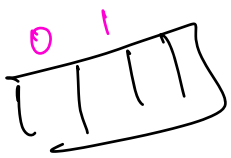
Goal:- Sort the array

A: [6, 5, 3, 2, 8, 10, 9] , K = 3

(0 1 2 3 4 5 6)

Sorted A:- [2, 3, 5, 6, 8, 9, 10]





0 → 1 ⟵ 2
0 → 2 ⟵ 3
0 → 3 ⟵ 4
0 → k ⟵ (k+1) elements

- pick (k+1) element.
- get min.
- place it at (5)[0] → new sorted array.

3-3 to 3+3
0 to 6

A: [0, 1, 2, 3, 4, 5, 6] , K = 3

max(0, 4-3)
min(4+3, 6)

KTH = (4)

C: [2, 3, 5, 6, _, _, _]
0 1 2 3 4

(1, b)

2+3 → (5)

max(2-3, 0) → 0

4th min.

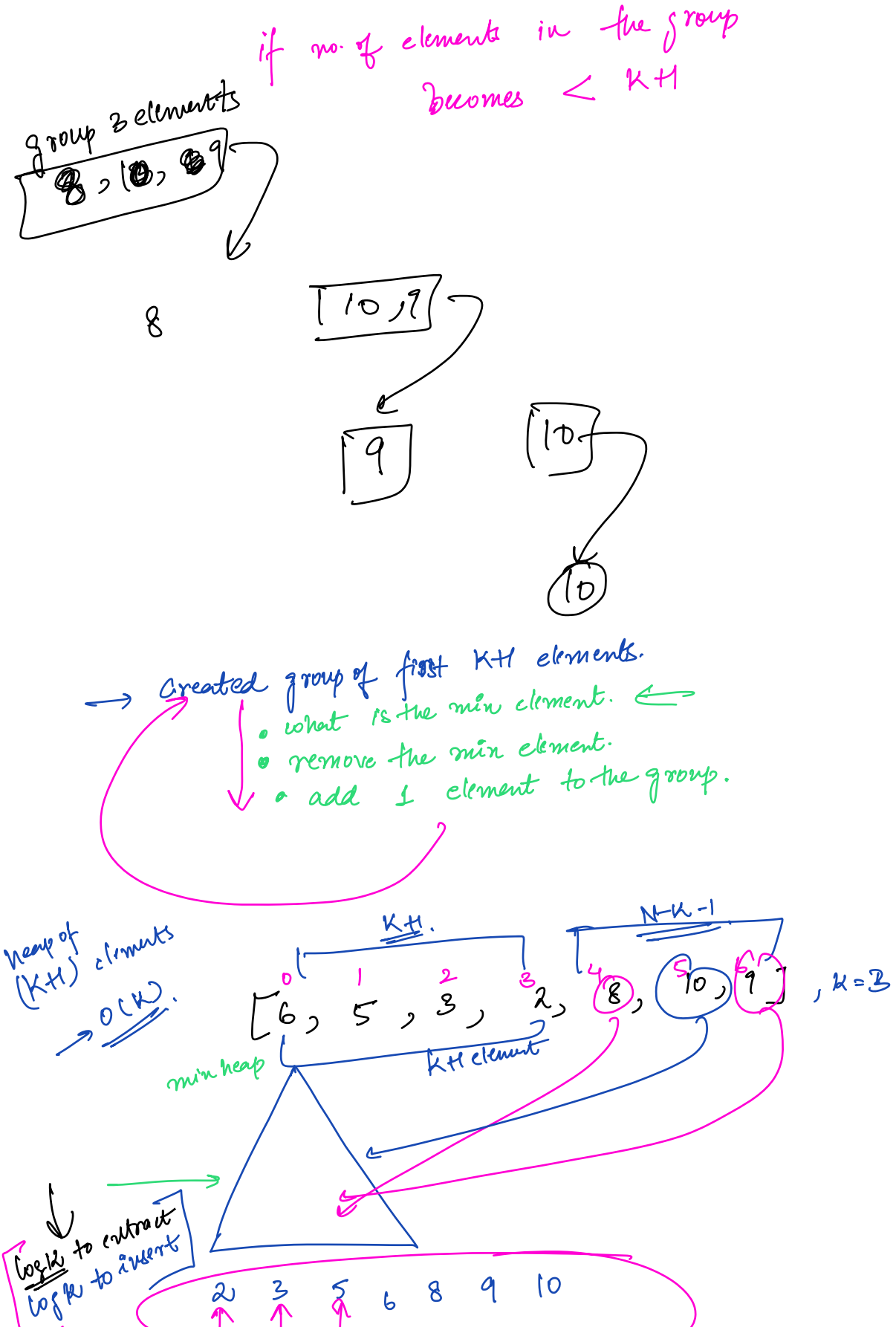
5th min

1+3 → 1, 2, 3, (4)

0, 1, 2, X, 4

what is the min element?

0, 1, 2, 5, 4, 5 ⟵ 3rd min. element (index 2)


if no. of elements in the group becomes < K+1
group 3 elements
8, 10, 9
8
10, 9
9
10
10
Created group of first K+1 elements.
• what is the min element.
• remove the min element.
• add 1 element to the group.
heap of (K+1) elements
O(K).
K+1.
N-K-1
0 1 2 3 4 5 6
6, 5, 3, 2, 8, 10, 9, K=3
min heap
K+1 element
log K to extract
log K to insert
2 3 5 6 8 9 10

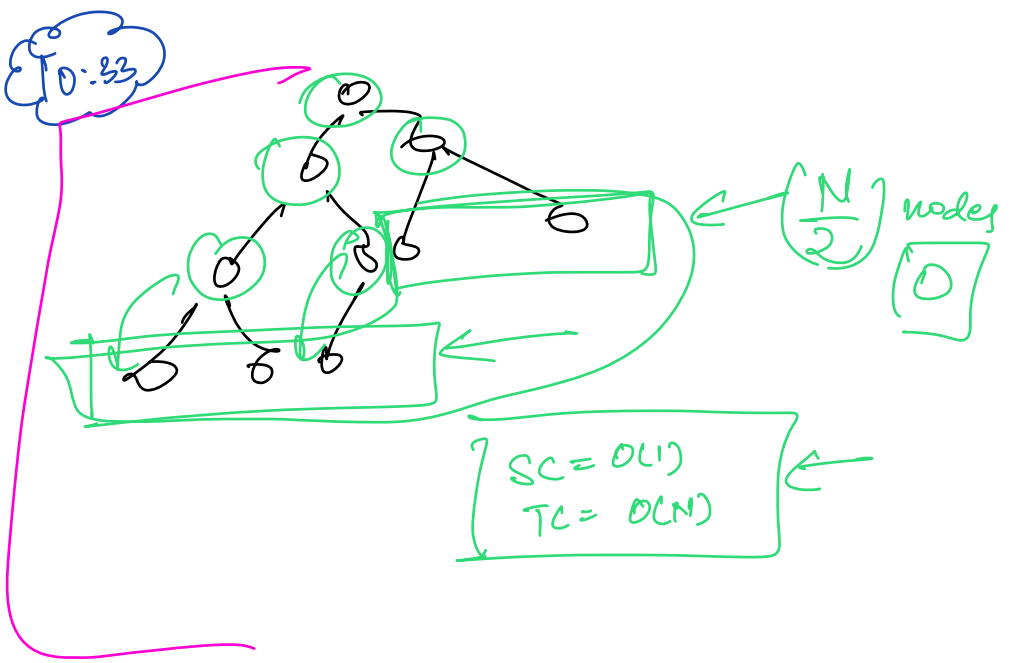
min heap of k+1 elements → O(k)

$(N-k-1) * \log k$

$TC = O(k) + (N-k-1)\log k$

$SC = O(k)$

↳ size of the heap.



Q. Given an array, sort it in place.

↳ S.C = O(1)

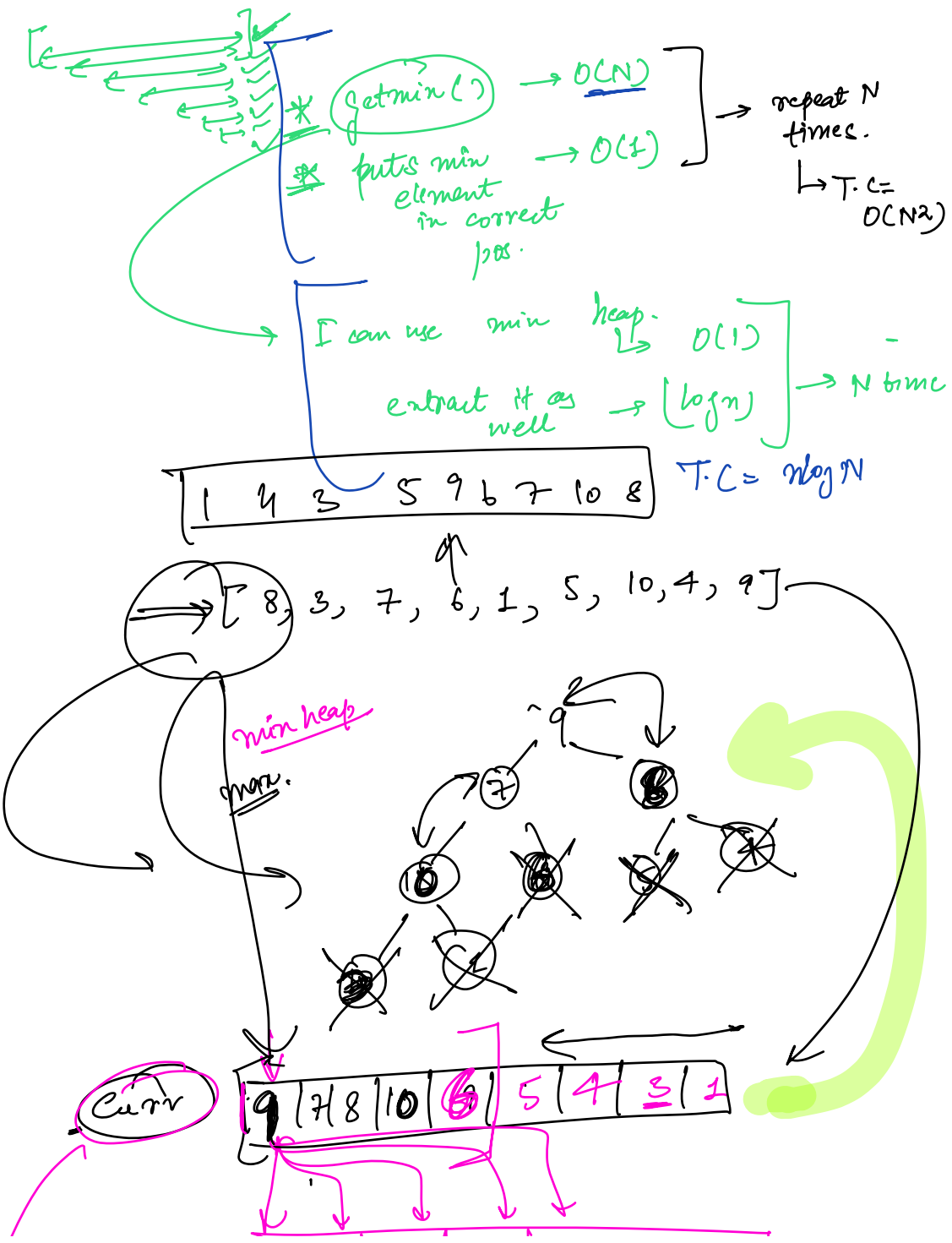
Insertion Sort.
Bubble Sort
Selection Sort

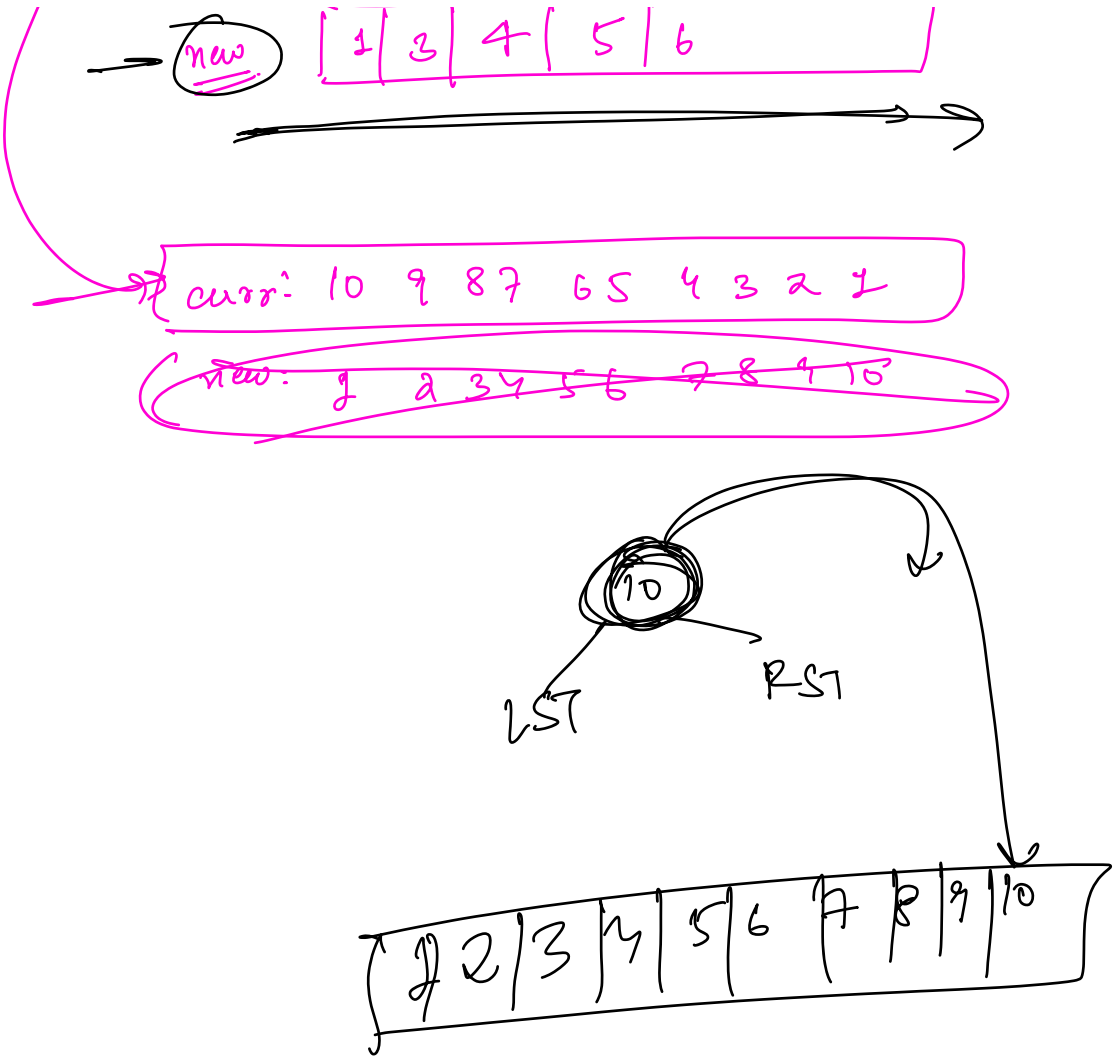
→ $O(N^2)$ ← T.C

S.C = O(1)

S.C = O(1)

T.C = O(nlogn)





Q Running median.

Given a stream of integers, find the median with every new insertion.

median.

9 → 9

9, 6 → 7

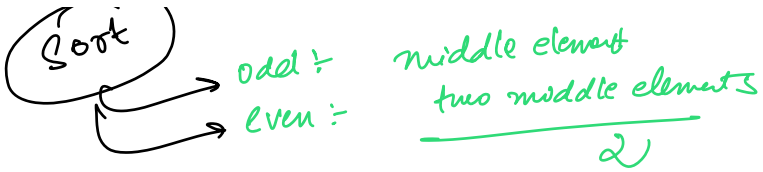
$\frac{9+6}{2} = \frac{15}{2} = 7$

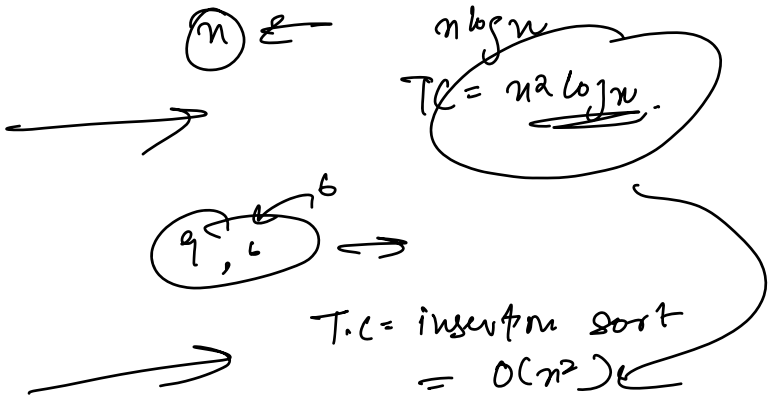
9, 6, 3 → 6

$\frac{9+6+1}{3} = \frac{16}{3} = 5$

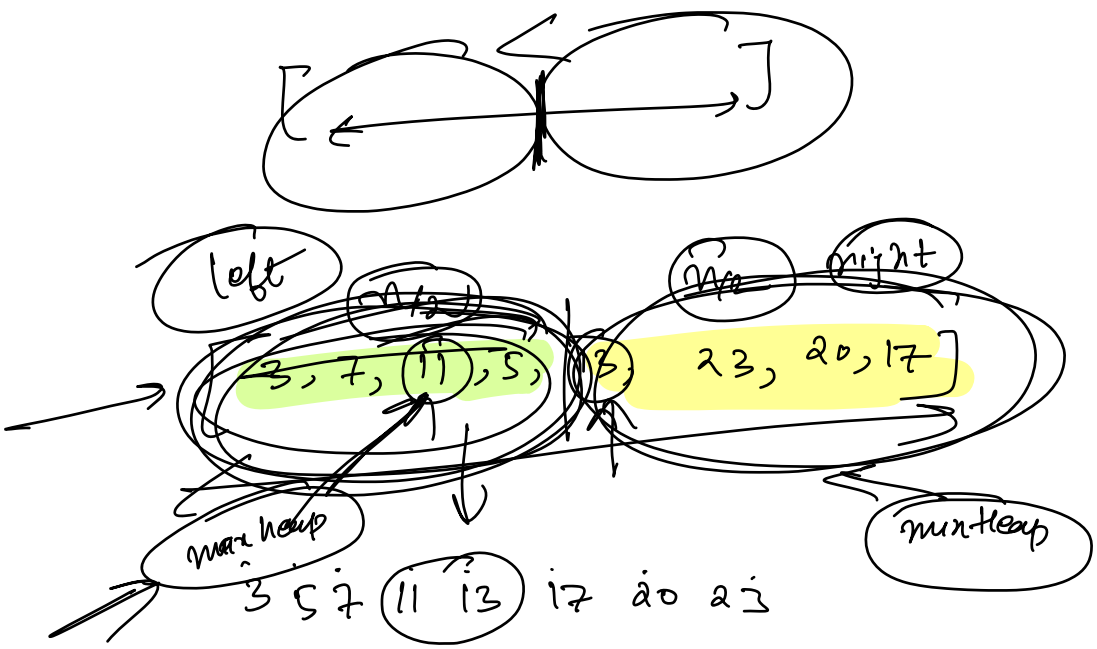
9, 6, 3, 10 → 7

3, 6, 9, 10 → $\frac{6+9}{2} = \frac{15}{2} = 7.5$



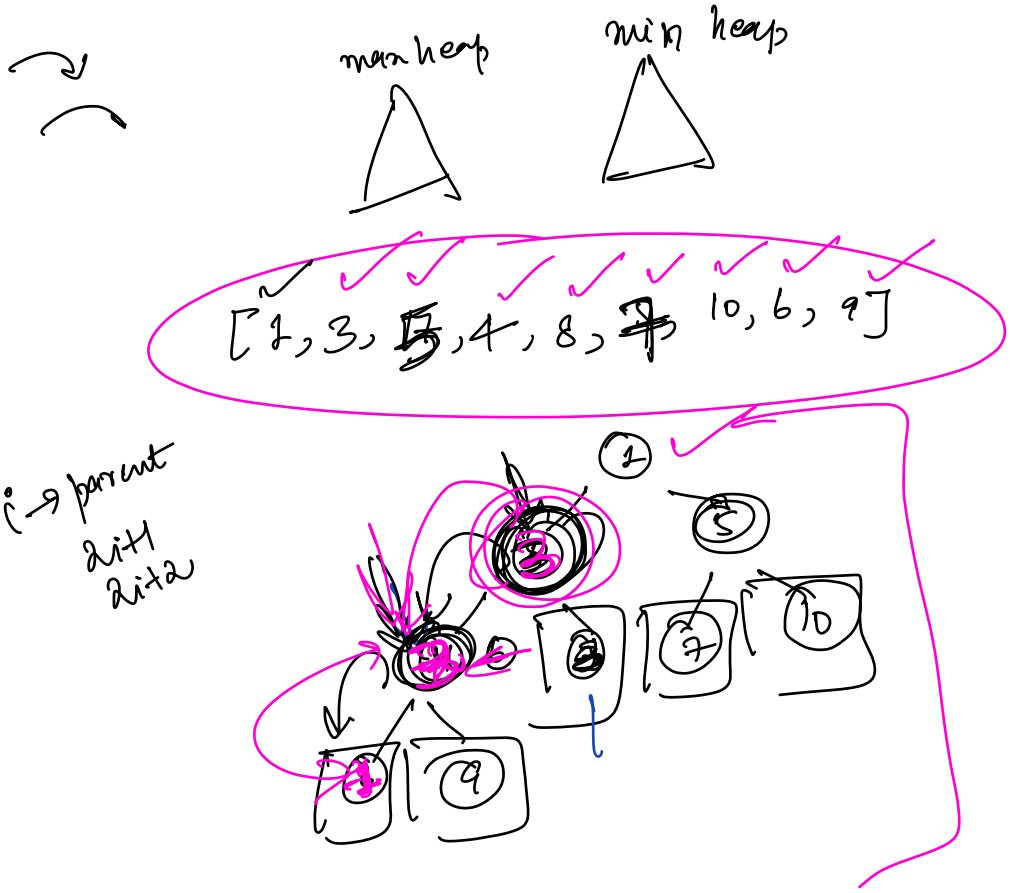


Do we need to sort the whole array to figure out median?

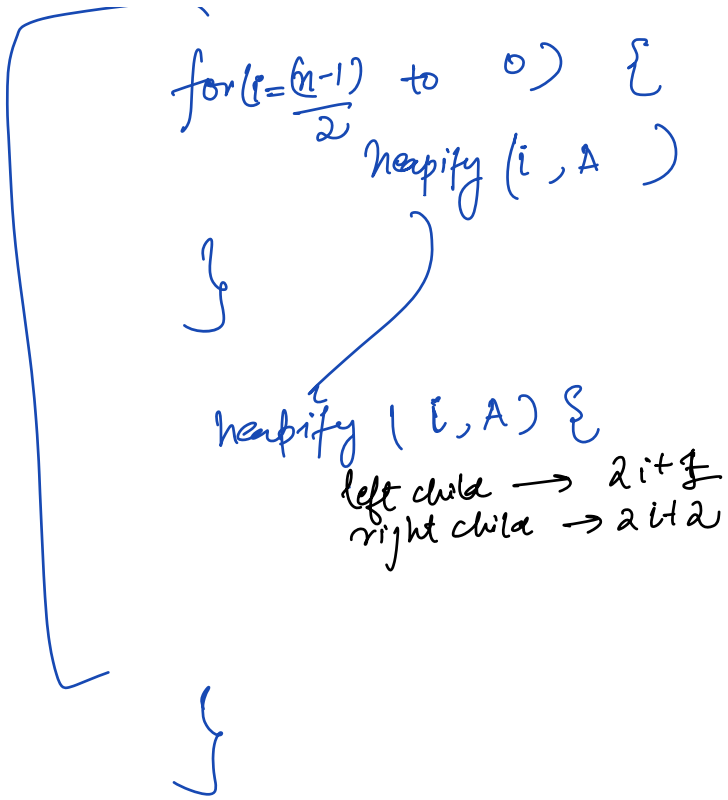


left group: If the array was sorted, these are first half elements

right group: If the array was sorted, these are the second half elements



index of the first non leaf node in a Complete Binary Tree → $\frac{(N-1)}{2}$


for(i=(n-1)/2 to 0) {
heapify(i, A)
}
heapify(i, A) {
left child → 2i+1
right child → 2i+2
}