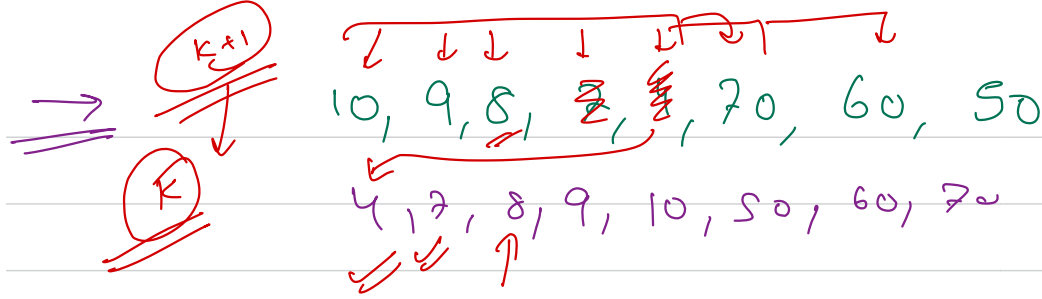


Q.2 Given an unsorted array, with special property.  
that it is not completely sorted. Instead it is  
K-sorted meaning every element in the array  
is at a distance at most K from its sorted  
position. Your task is to sort the array in less  
than  $O(n \log n)$ .  $a[i] \leq 10^9$

10, 9, 8, 7, 4, 20, 60, 50

K = 4

4, 7, 8, 9, 10, 50, 60, 20



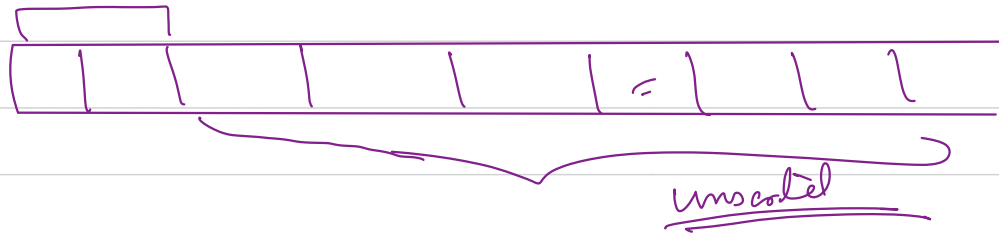
Every time we just need to maintain  $K+1$  elements. And the minimum of those  $(K+1)$  elements will be the correct element at the

1<sup>st</sup> iteration

$(K+1)$  size heap (min heap)

$$(K+1) \log K + (n-K-1) \log K \rightarrow \underline{\underline{n \log K}} \quad \underline{\underline{O(n \log K)}}$$

empty  
sorted  
region



$$TC \underline{O(n^2)}$$

$$SC \rightarrow \underline{O(1)}$$

Is selection sort stable?

→ least no. of swaps

a) yes

b) No XXXX

Heap Sort

4' 2 3 4' 1  
1 2 3 4'' 4'

No

Instead of swapping, can we do shifting / insertion

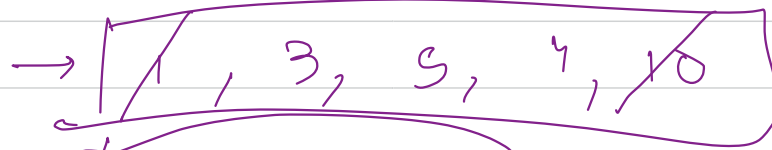


min heap

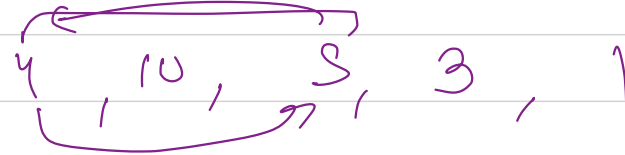
$O(n)$

$scan = n-1$

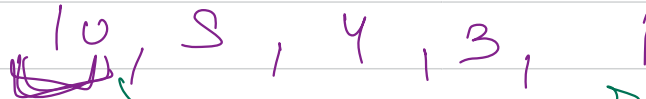
Heap Sort



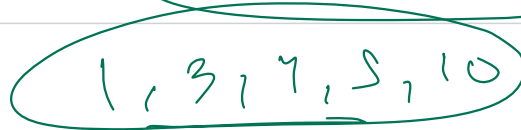
$scan = n-2$



$scan = n-3$

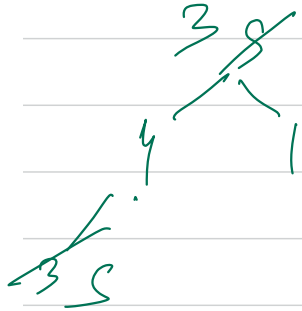


$scan = n-4$



$scan = n-5$

4, 10, 5, 3, 1



10, 4, 5, 3, 1

5, 4, 1, 3, 10

3, 4, 1, 5, 10



sorted

$O(n)$

TC  $\rightarrow O(n + n \log n)$   
 $\rightarrow O(n \log n)$

SC  $\rightarrow O(1)$

$$\underline{\underline{O(n \log n)}}$$

(1) Unstability

(2) The constant factors <sup>and lower order terms</sup> of the heap sort that are avoided in big O notation are quite big and these constant terms & lower order factor are greater than merge and quick sort.

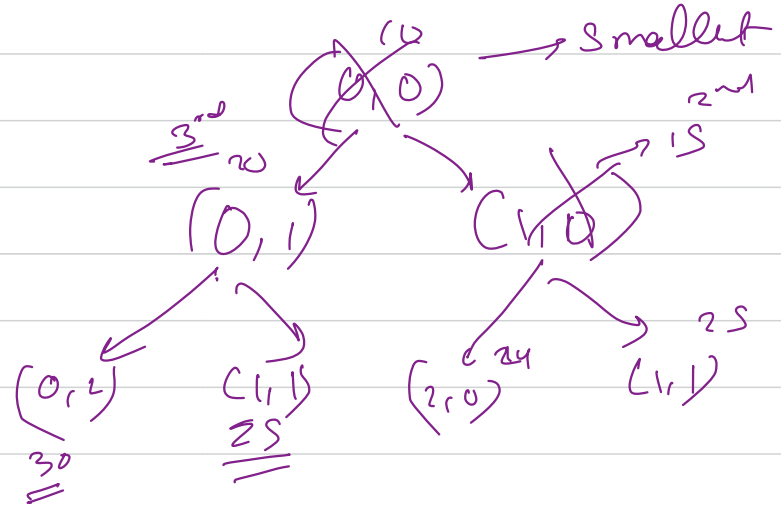
Q.1 Given a row wise and column wise sorted 2D array. find the  $k^{\text{th}}$  smallest element.

$\begin{matrix} 10, 20, 30, 40 \\ 15, 25, 35, 45 \\ 24, 29, 32, 48 \\ 32, 33, 39, 50 \end{matrix}$

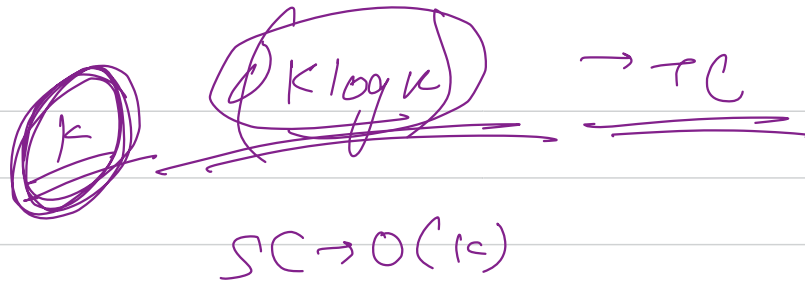
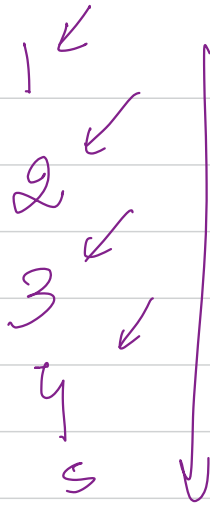
$k=2 \rightarrow \text{ans} \rightarrow 15$

Some present  
code  $\rightarrow S$   
 $10, 20, 50$

$(i, j) \rightarrow S - (i, j)$   
 $+ (i+1, j) + (i, j+1)$



Set of  
Min heap





Q - You are given a grid of row wise sorted elements. find the  $k^{\text{th}}$  smallest.

$\begin{bmatrix} 2 \\ 10 \\ 22 \end{bmatrix}$

1, 2, 3, 4

10, 20, 30, 40

-1, 0, 22, 33

$O(k \log n)$

$n \rightarrow \text{size of heap}$

rows  
↓  
1 x m x n

$-1, 0, 1$   
 $\swarrow$   
1<sup>st</sup> smallest

$\begin{bmatrix} i \\ j \\ val \end{bmatrix}$

$(i, j) \rightarrow (i, j+1)$   
 $S = (i, j) + (i, j+1)$

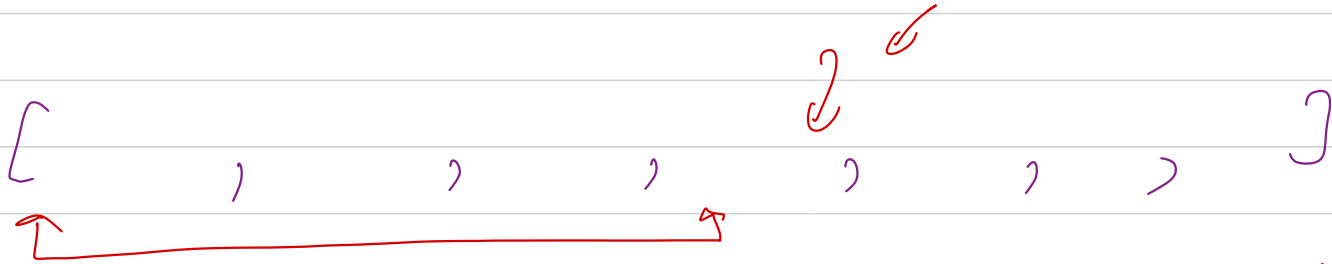
Q<sup>n</sup>

Given  $K$  sorted arrays. Merge them  
in a single sorted array. Each array  
can be of max size  $n$



→ sort →  $k^{\text{th}}$  element →  $n \log n$

/// build heap of whole array & pop  $k$  elements  
→  $O(n \log n)$



max heap

new elem < top of heap

$n \geq k$

→ max heap

→ top of heap

$k < n$

$TC \rightarrow O(n \log k)$

$SC \rightarrow O(k)$

Check for the first unk of size k & create

a heap (max heap) of size k

2 2 2

3 odd elements

3, 1, 14, -1, 2, 6

2

2, 1, -1



→ k smaller  
ele