K^th Largest Term

Heuristic: "K" => Priority Queue

[3 2 3 1 2 4 5 5 6]

natural order

$\overset{3}{a} < \overset{5}{b}$	negative	a-b
$\underset{3}{a} = \underset{3}{b}$	0	a-b
$\underset{5}{a} > \underset{3}{b}$	positiv	a-b

Approach 1: Sort Descending
return K^th term

Arrays.sort(nums, (a,b) -> b-a)
T.C. O(nlgn)

reverse order (opposite)

$\overset{3}{a} < \overset{5}{b}$	b-a
a = b	b-a
$\underset{5}{a} > \underset{3}{b}$	b-a

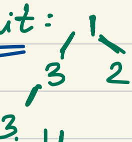
Approach 2:
put all elements in MaxHeap
pop K times.
→ nlgn + klgn = O(nlgn) No Better

Approach 3:
We still need to iterate on all elements (First Principles)
Can we limit heap to K; as that's what we are looking for.

Loop Invariant: At any given time heap holds K largest elements so far.

~~3 2 3 1~~ 2 4 5 5 6

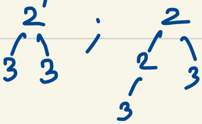
Init:



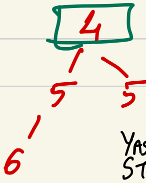
min Heap

Why minHeap:
Any any given time you can pop Min if you get good element

Loop
element = 2
pop: 1, add(2) ...



Termination



Result.

Yash Pradhan Study Notes