Heap

DSA

Easy - Intermediate

80%

- Array & Strings
- Linked List
- Stack & Queue
- Recursion
- Hashmap

Intermediate - Advance

20%

- Tree
- Graph
- DP
- Segment tree
- Tries

Honest

Complete Binary Tree

- Minheap
- Maxheap

Heap Data Structure

- Kth minimum element
- Kth largest element

Merge Sort → O(nlogn)

Input: nums = [3,2,1,5,6,4], k = 2
Output: 5

Sort

Approach 1

n = 6

0	1	2	3	4	5
1	2	3	4	5	6

n-k → index number

O(1)

⇒ O(n log n)

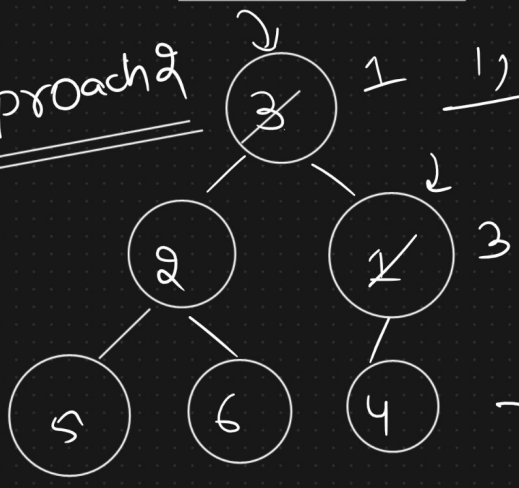
Space complexity → O(n)

Input: nums = [3,2,1,5,6,4], k = 2
Output: 5

kth Largest element

Minheap

Approach2



Maxheap

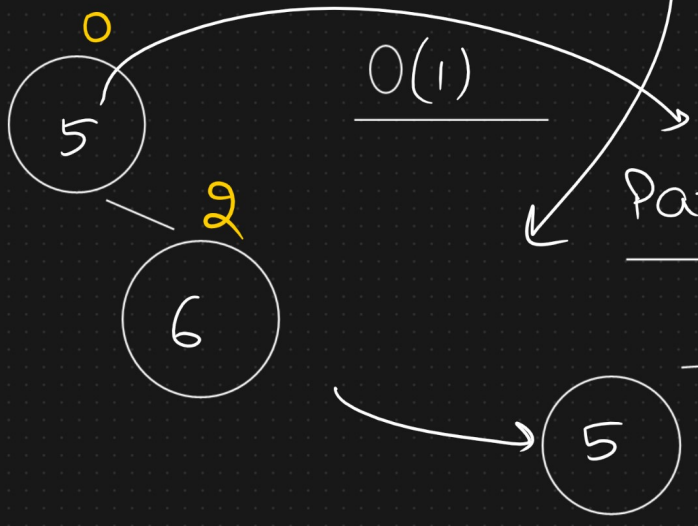
Approach3

$(n-k) \cdot \log_2 n$

3 times deletion from minheap

$(n-k)$ items

$(6-2) = 4$



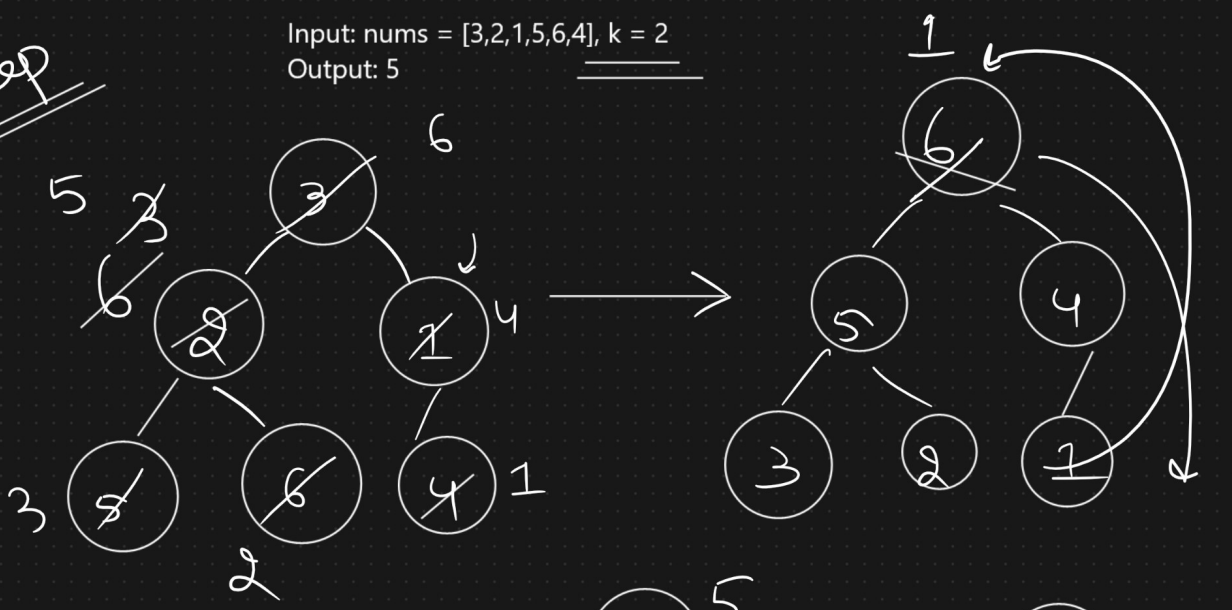
O(1)

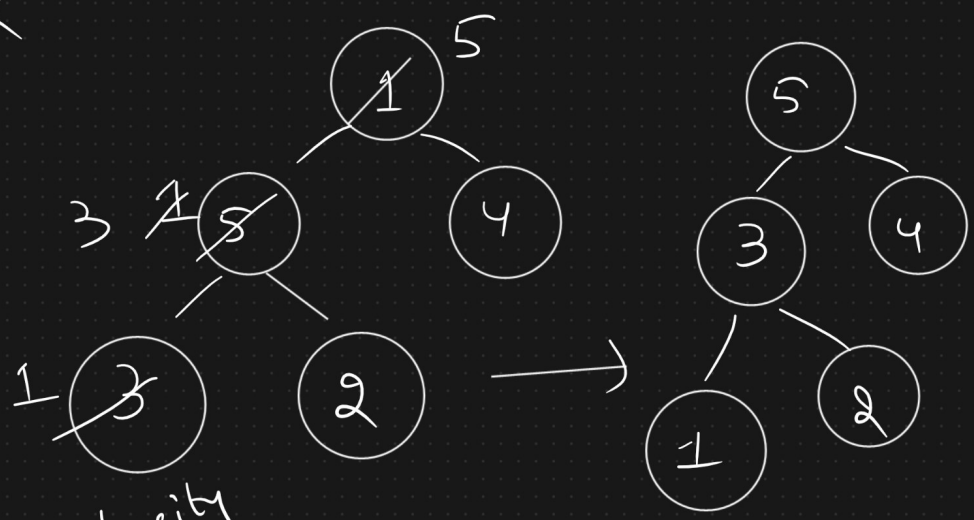
Parent/Root node

Space complexity → O(1)

Input: nums = [3,2,1,5,6,4], k = 2
Output: 5

Maxheap



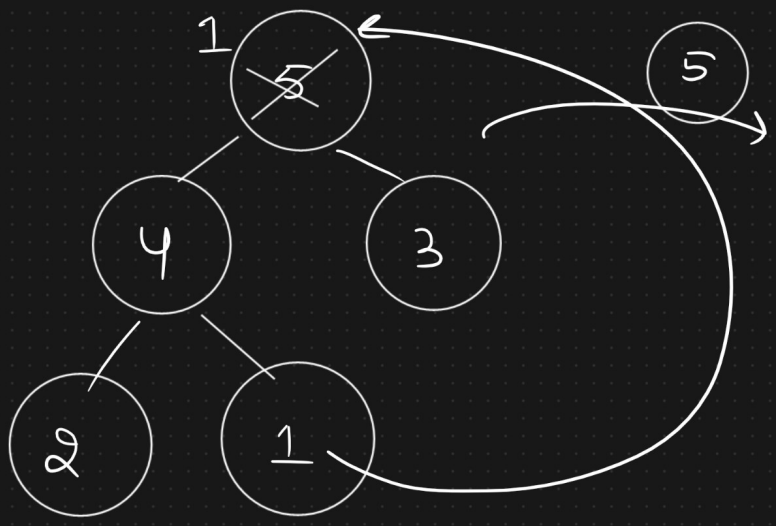


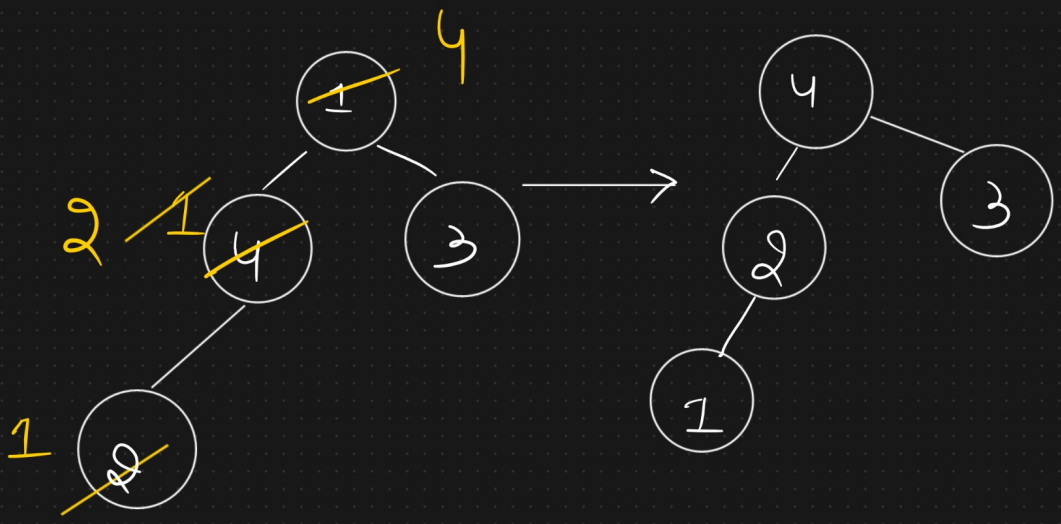
time complexity

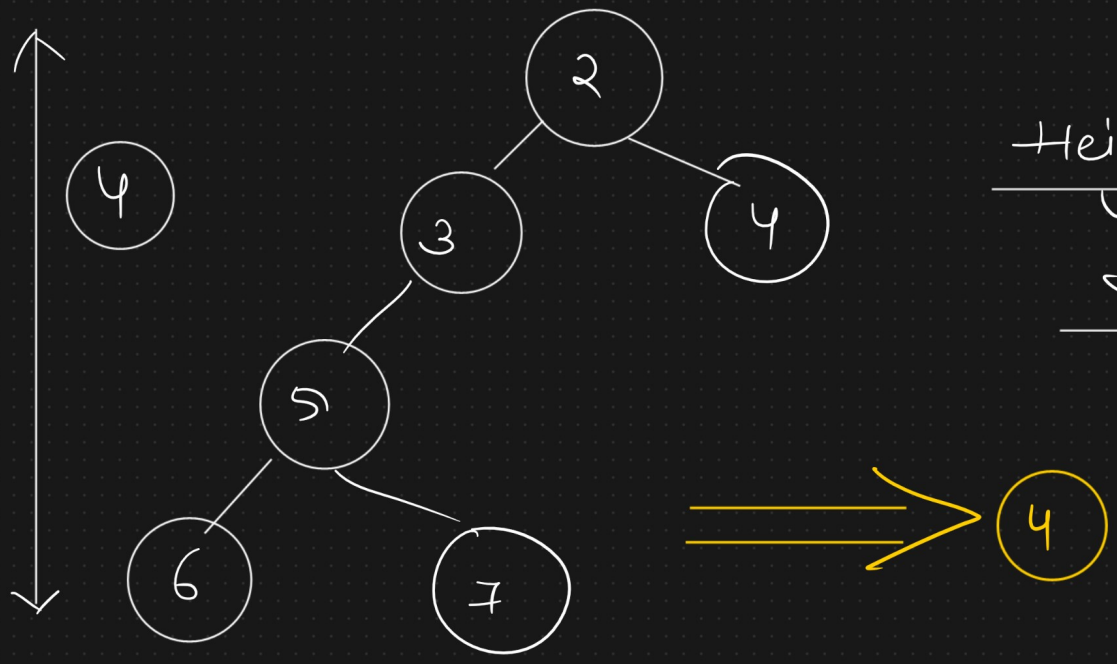
$(K-1)\log n \approx K \log n$

Space complexity

$\hookrightarrow O(1)$


1
5
5
4
3
2
1


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



Height = Depth