

$[a_0 \ a_1 \ a_2 \ a_3 \ \dots \ a_{n-1}]_n \rightarrow \underline{\underline{n \leq 2 \times 10^5}}$

pair that has max possible xor value.

\rightarrow (array of size n) \rightarrow lot of pairs are possible

\rightarrow approx n^2 \rightarrow TLE

XOR



a

0

0

1

1

b

0

1

0

1

$a \wedge b$

0

1

1

0

[3, 10, 5, 25, 2, 8]



→ array

3	→	0	0	0	1	1
10	→	0	1	0	1	0
5	→	0	0	1	0	1
25	→	1	1	0	0	1
2	→	0	0	0	1	0
8	→	0	1	0	0	0

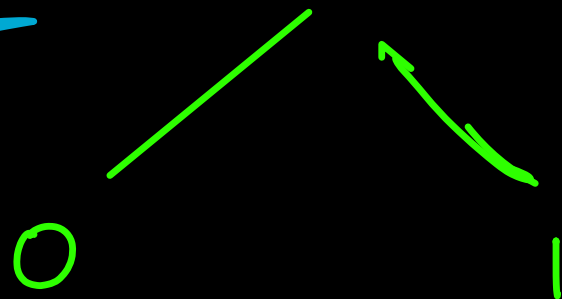
25
3 10 5

(5 bits) $x \wedge y \rightarrow 11111$

→ common prefix
 → msb until
 prefix

→ max xor pair

Binary Trie



3, 10, 5, 25, 2, 8

$$J_i$$

→ root

ans = 0

28

0 0 0 1 0

11001

1 1 0 1 1

$$16\tau + 8 + 2\tau)$$
$$26 + 1$$

27

$$O(n \times \underset{L}{\text{length of branch}})$$

32 bit

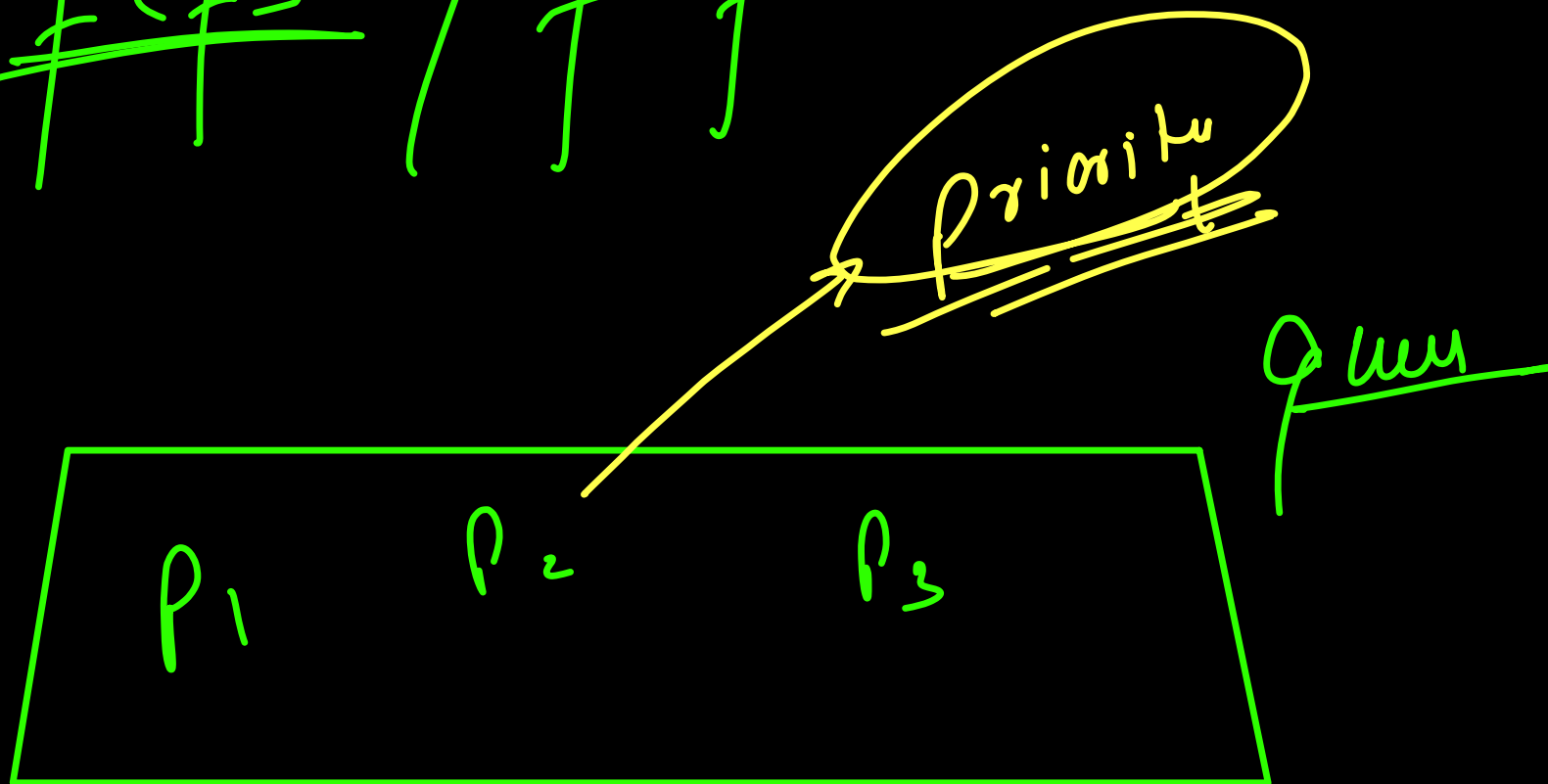
$$O(n \times 32)$$

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

(when an element is added)

Priority Queue

queue \rightarrow fcfs / fifo



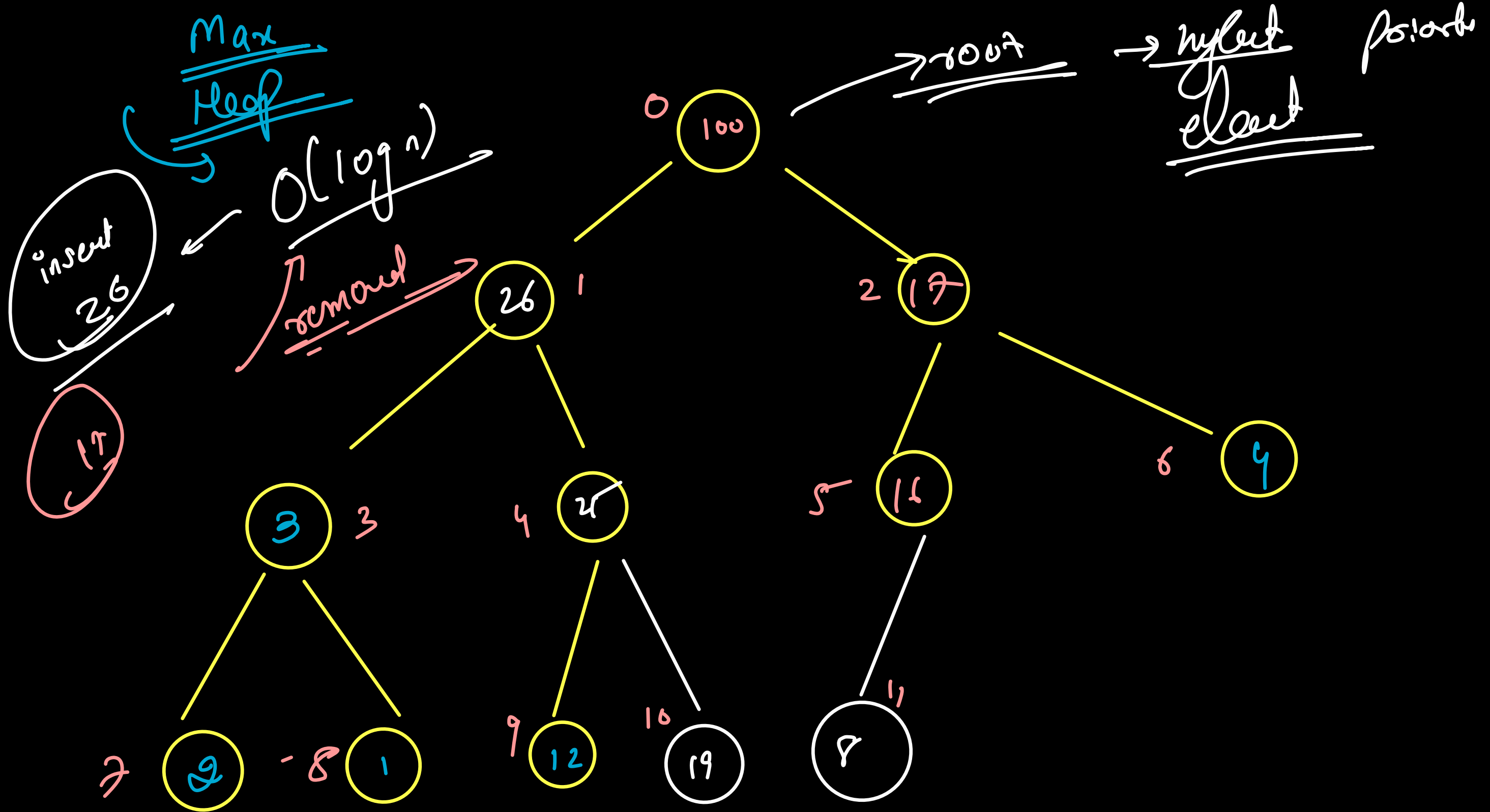
Tree data structure → Heap ^{implement} → pg

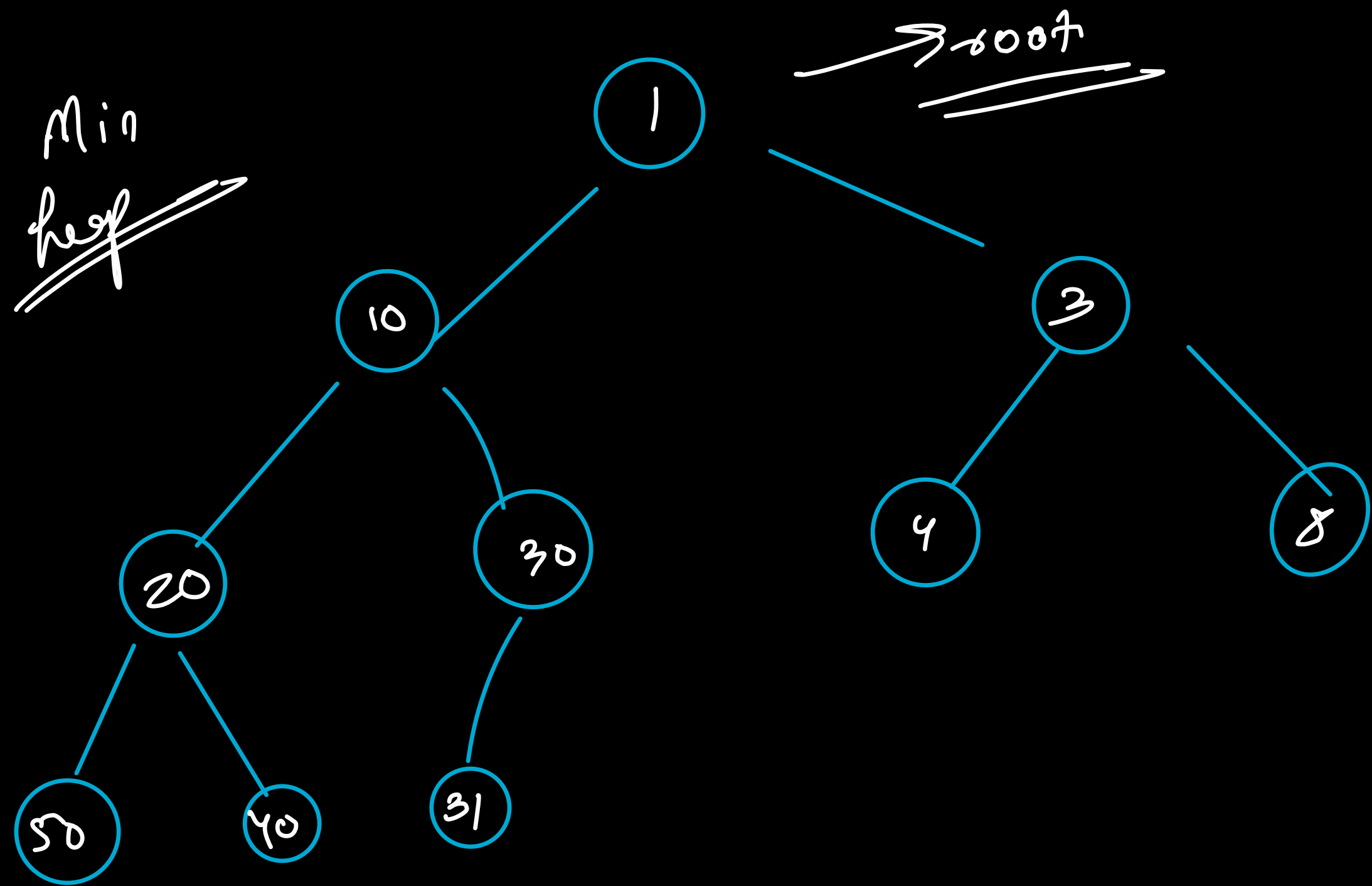
→ Binary Tree → Complete binary Tree

→ priority of parent nodes > priority of child nodes

→ n no. → priority → bigger

bigger the no higher the priority





$$p_i \rightarrow i \quad \rightarrow \quad \begin{aligned} lc &\rightarrow 2i+1 \\ rc &\rightarrow 2i+2 \end{aligned}$$

$$c_i \rightarrow i \quad \rightarrow \quad p_i \rightarrow \left\lfloor \frac{i-1}{2} \right\rfloor$$

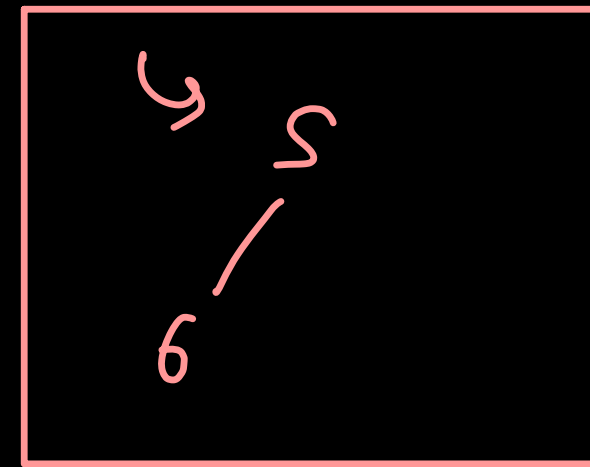
insertion \rightarrow upheapify

removal \Rightarrow down heapify

Size k → min heap

↓ ↓ ↓
3, 2, 1, 5, 6, 4

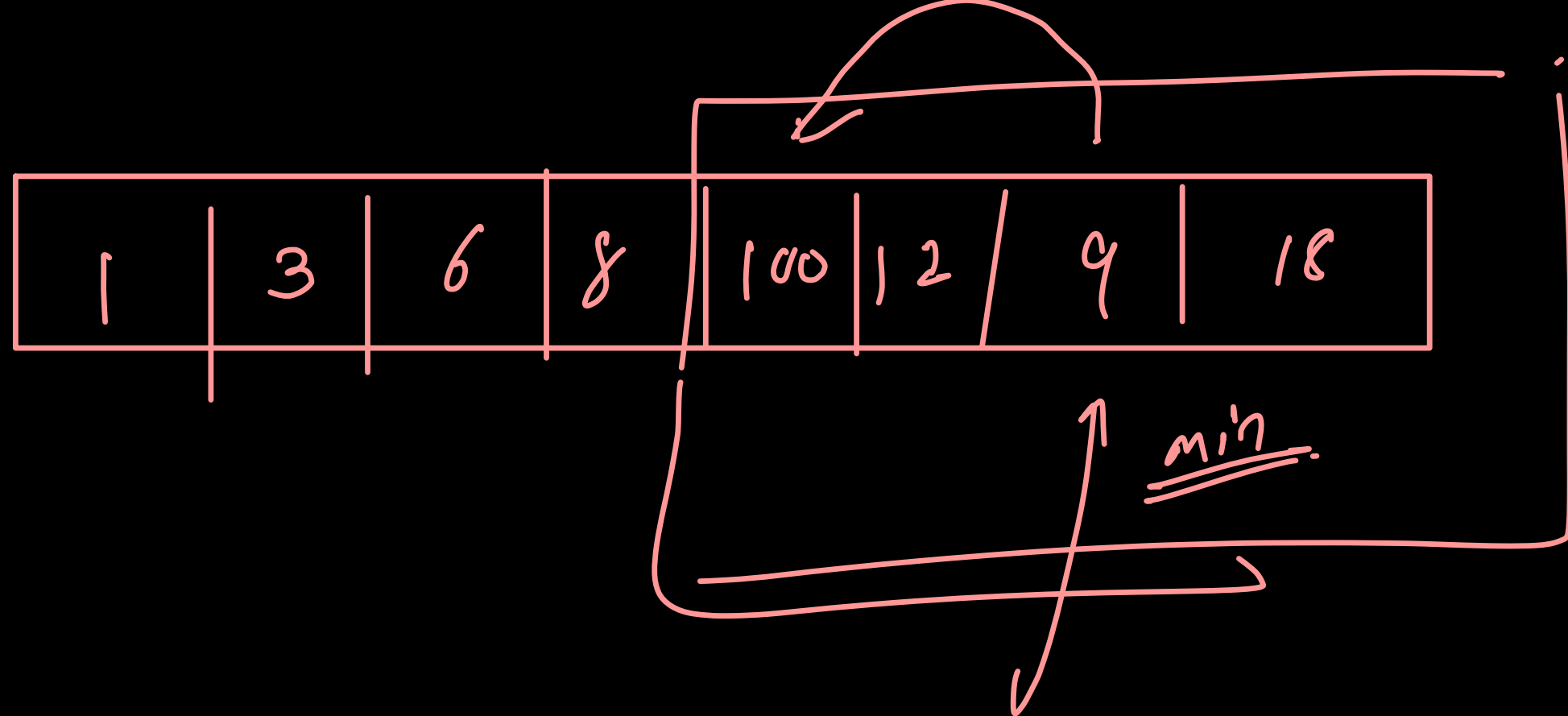
(kth largest
element)

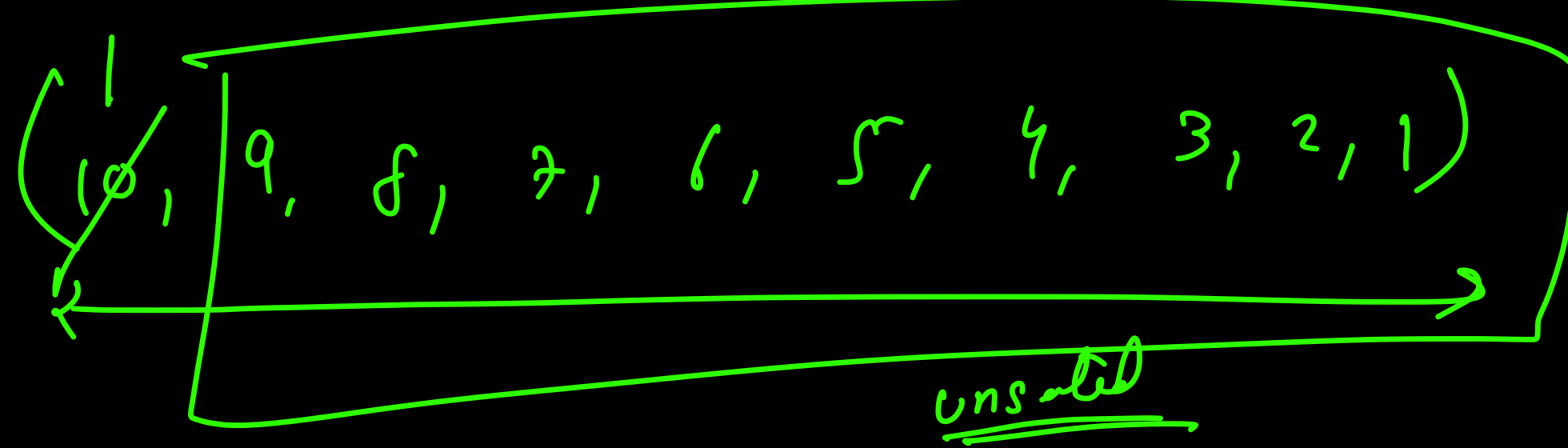


k=2

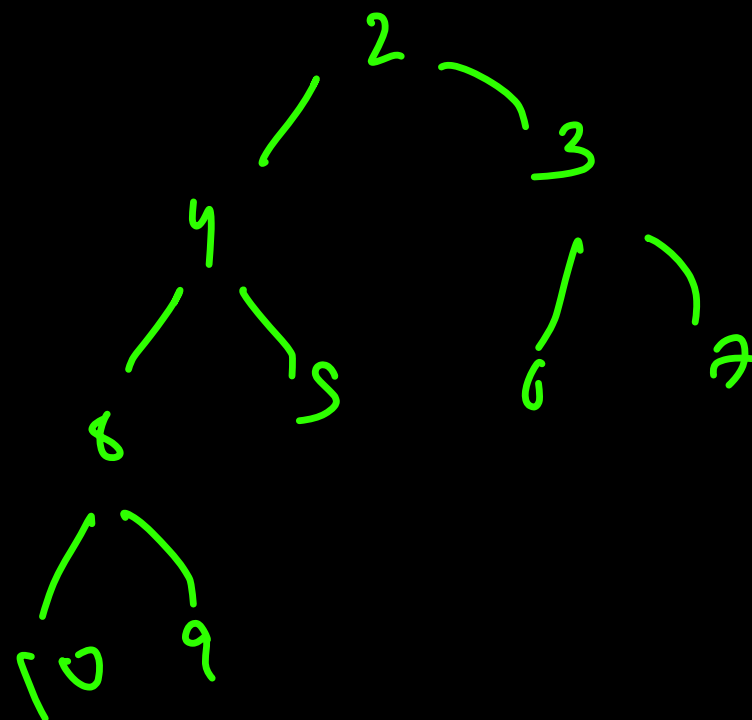
$O(n \log k)$

if we maintain k biggest
in a Min heap





$O(1)$



$$\rightarrow \frac{n \log n}{1} - n \log n \rightarrow \underline{\underline{O(n \log n)}}$$

merge Sort