Priority Queue

Important Ques

→ Buy the ticket

→ Running median

→ Inplace heap sort

→ k^{th} largest (using min and max heap).

k largest / k smallest

k^{th}

Priority Queue

↓

Queue

(i) insert
(ii) getMax / getMin
(iii) removeMax / removeMin

(i) Min Priority Queue

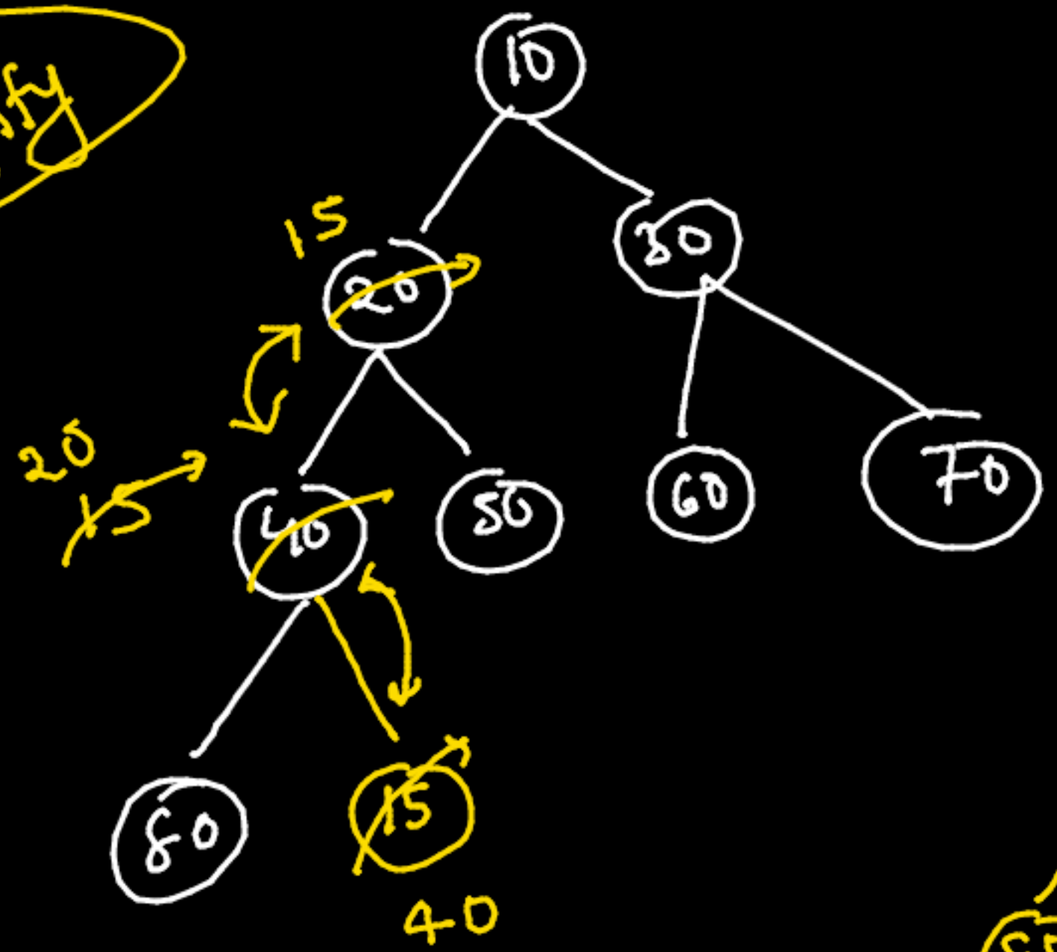
(ii) Max " "

priority_queue <int> pq; → Max heap

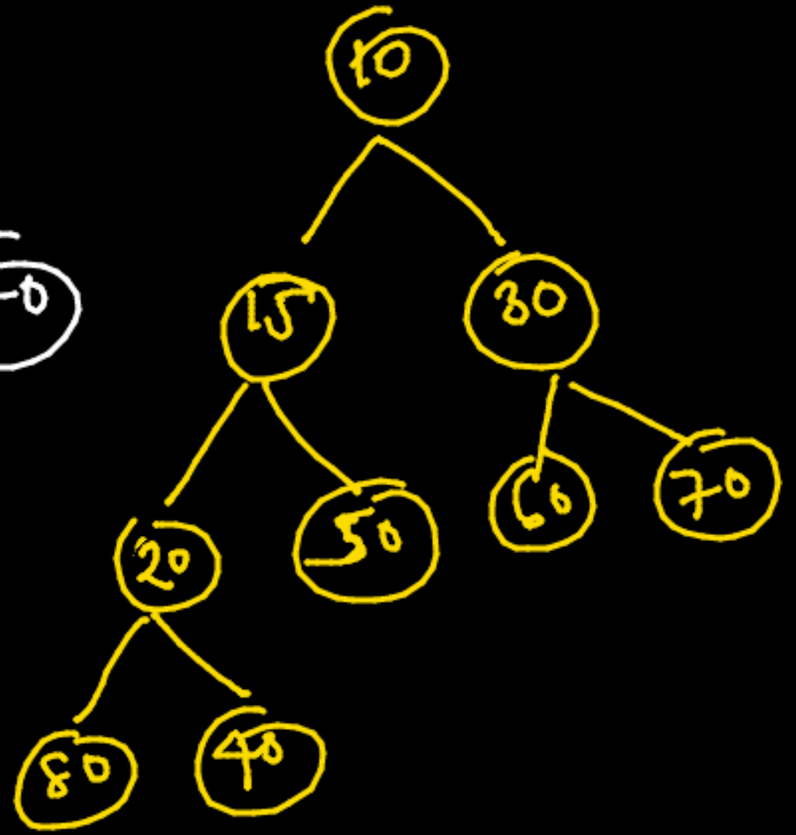
priority_queue <int, vector <int>, greater <int> > pq; → Min heap

Insert in Min Heap

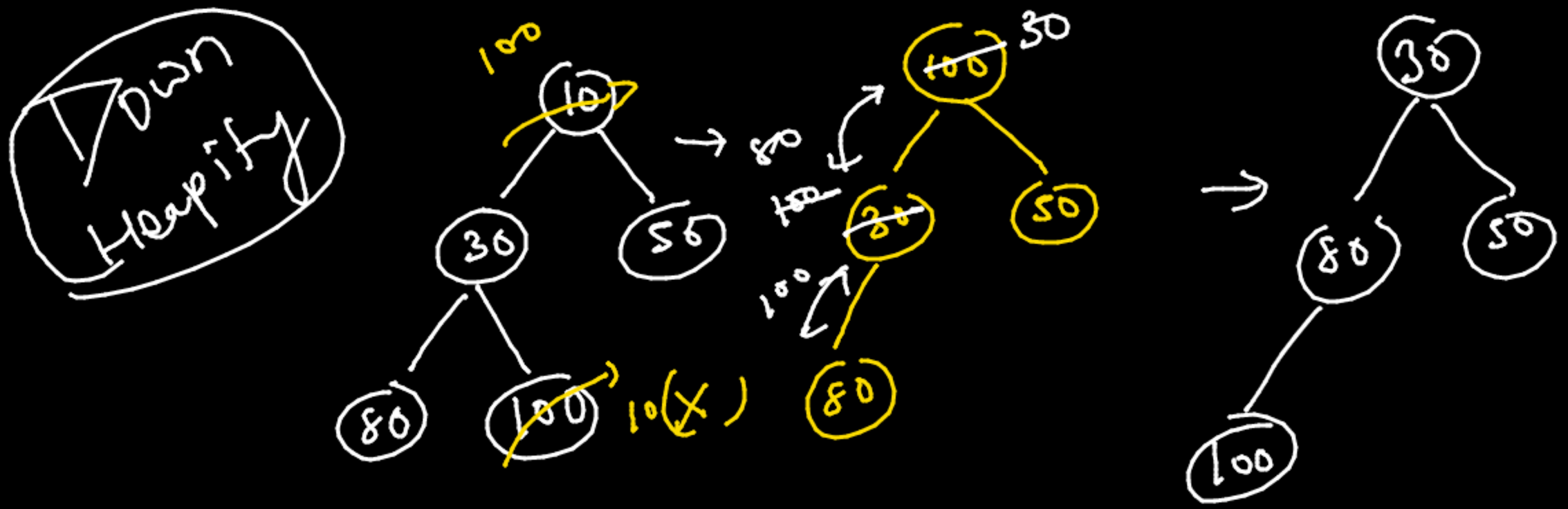
Up-Heapify



insert → 15.

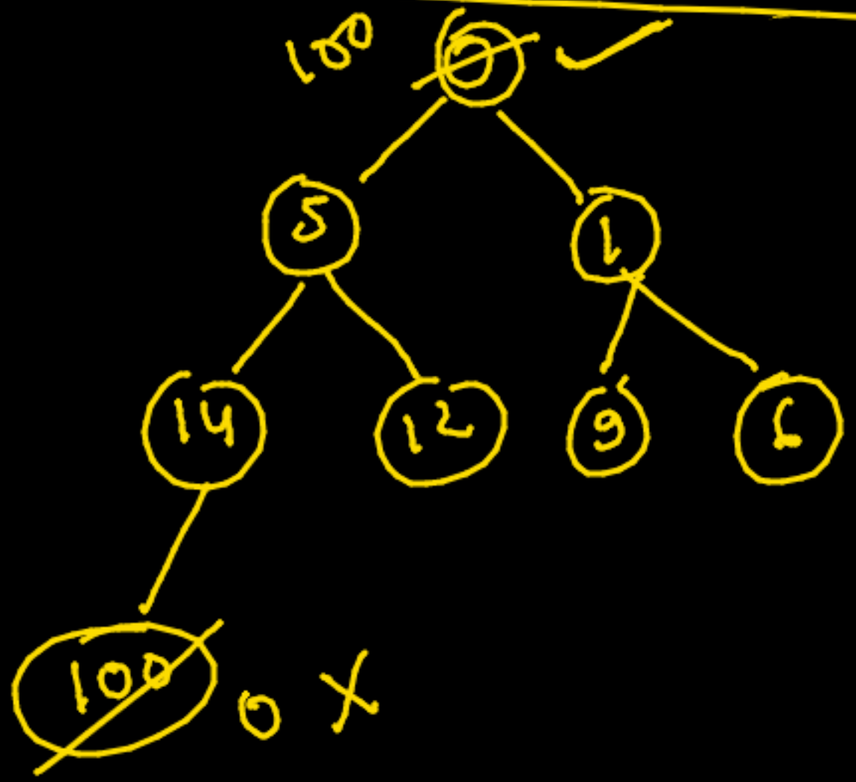


Delete In Min Heap



Make min Heap

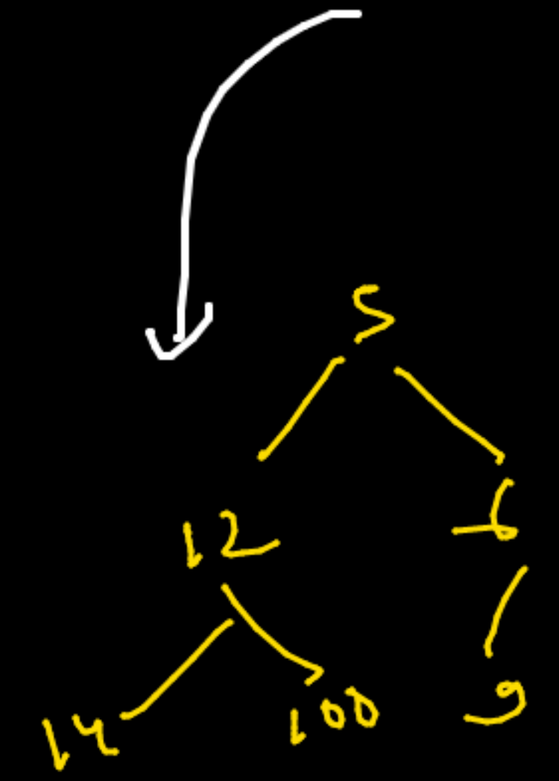
12, 6, 5, 100, 1, 9, 0, 14

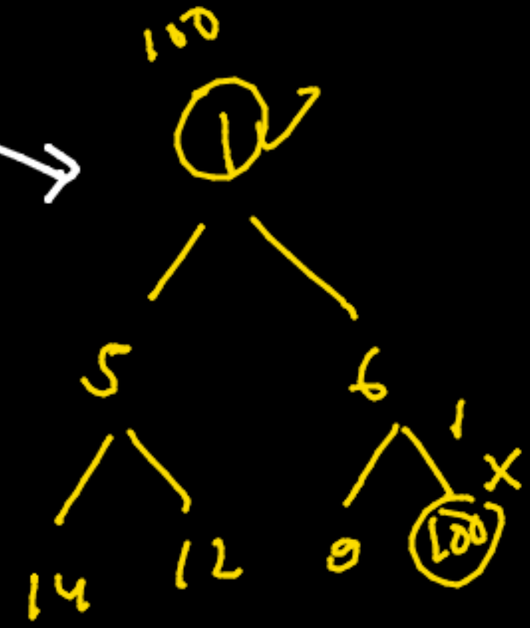


0

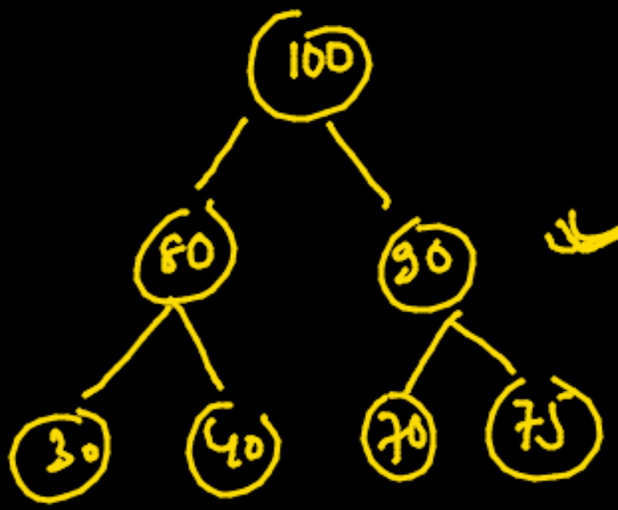
(0) Remove min()

(1) Remove min()





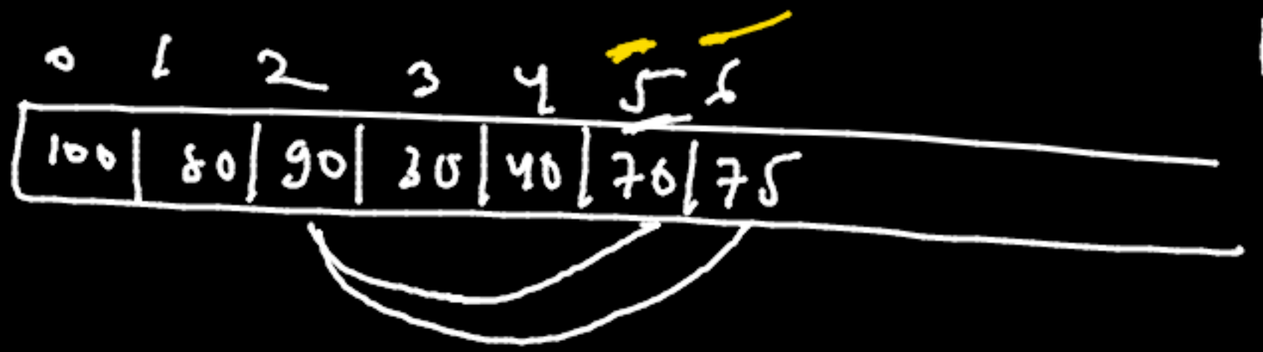
Check Max Heap



if parent < child
return false;

100 <

$\frac{6-1}{2} = 5/2$ (2)



parent (2) = $\frac{\text{child}-1}{2}$ (2)

$\frac{5-1}{2}$

k^{th} largest element

- Using max heap
- Using min heap.