Heap Data Structure

Minheap / Maxheap

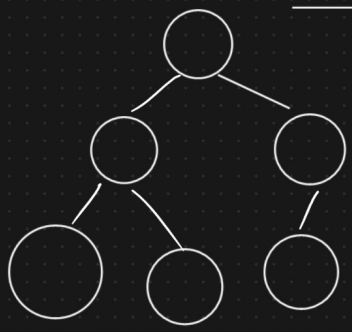
Complete Binary Tree

Atmost 2 child nodes

{ ① left side to Right side
② first level filling is completed → next level

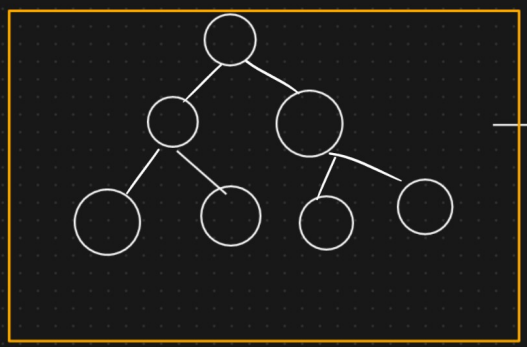
Almost complete Binary tree

1



full CBT ← strict form

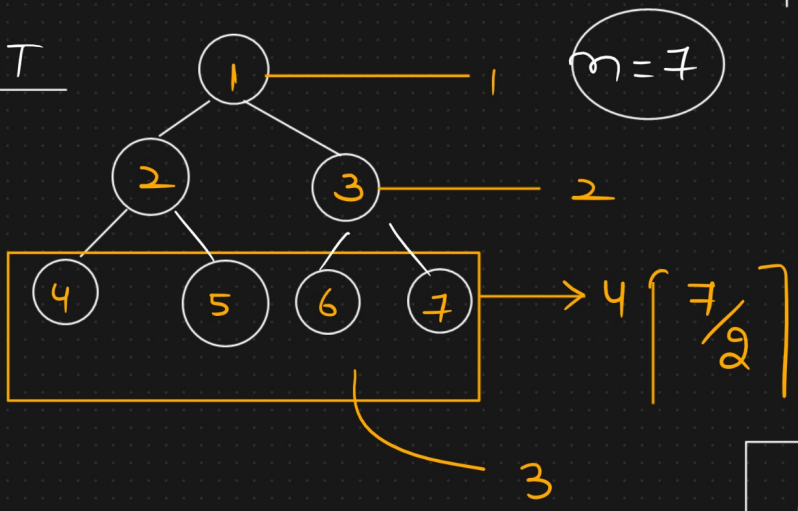
2



→ Number of nodes is greater

① Number of Leaf Nodes = $\lceil n/2 \rceil$

fCBT



k = # levels of CBT

k = 3

② $n = 2^k - 1$

$= 2^3 - 1$

$= 7$

③ k = ?? $2^k = n+1$

$k \log_2 2 = \log_2 (n+1)$

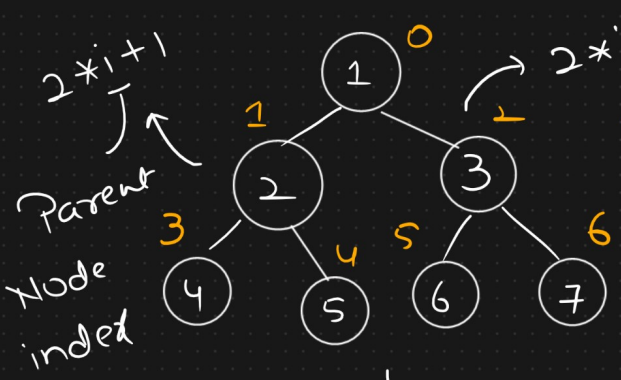
$k = \log_2 n$

Generalized form

↓

CBT

FCBT



Linked list

OR

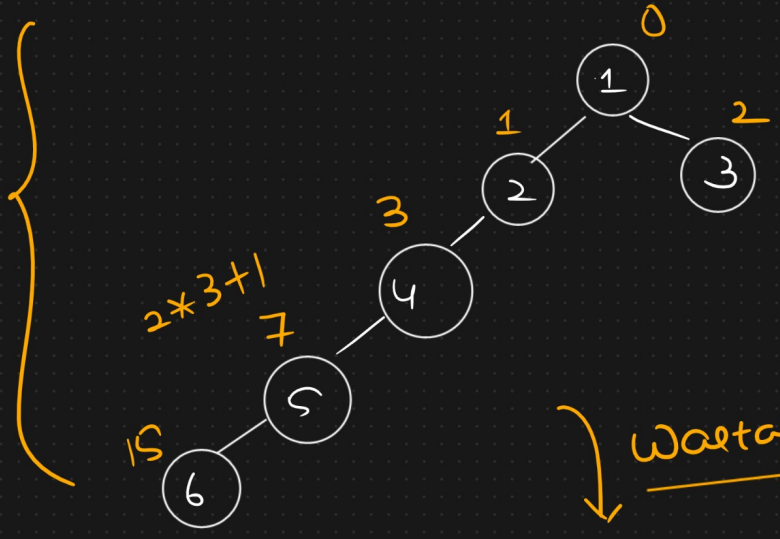
Array ✓

↳ Not wastage of Space

Array

0	1	2	3	4	5	6
1	2	3	4	5	6	7

Skewed Binary Tree



Linked list

Wastage ≠ space

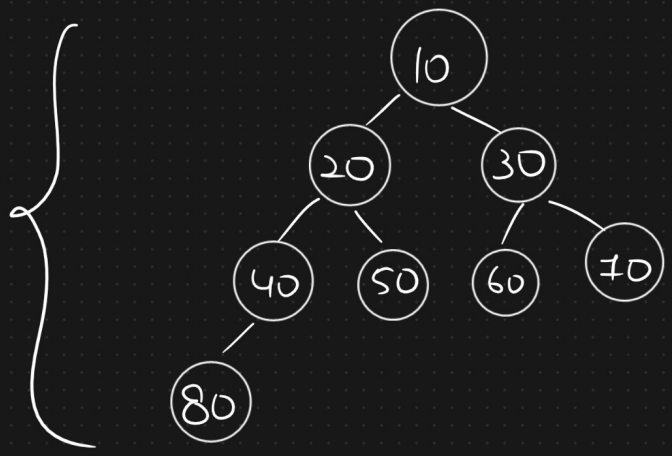
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4				5								6

Linked list

Minheap → Parent Node <= child node

- Insertion ✓
- Deletion
- creation ✓
  - Heapify

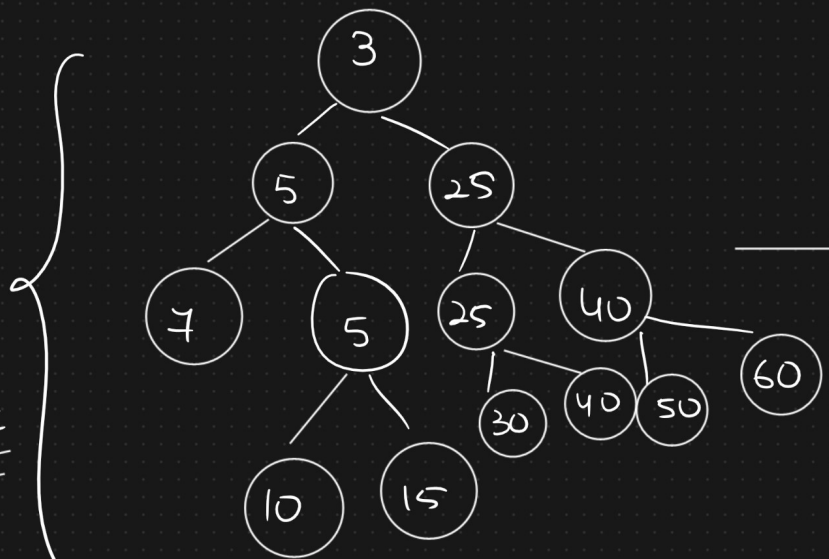
10, 20, 30, 40, 50, 60, 70, 80



Minheap ??

1. CBT
2. Parent <= child node value

GATE



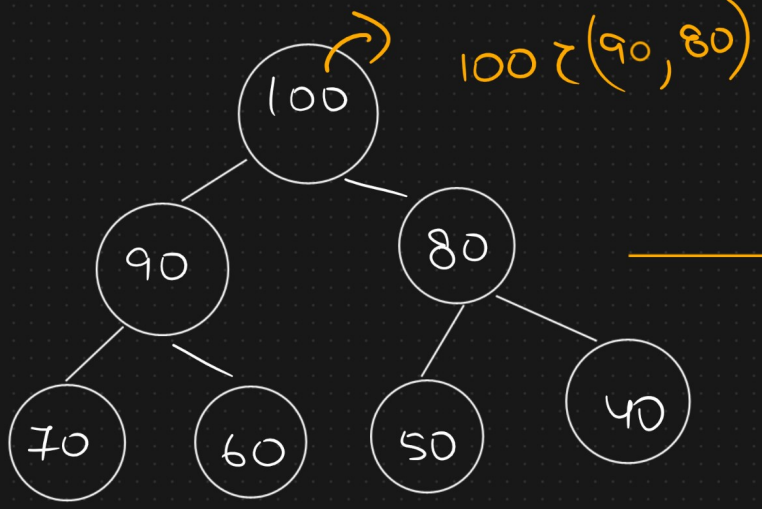
→ Not a Minheap

- CBT X

Maxheap

1. CBT
2. Parent node >= child node

100, 90, 80, 70, 60, 50, 40

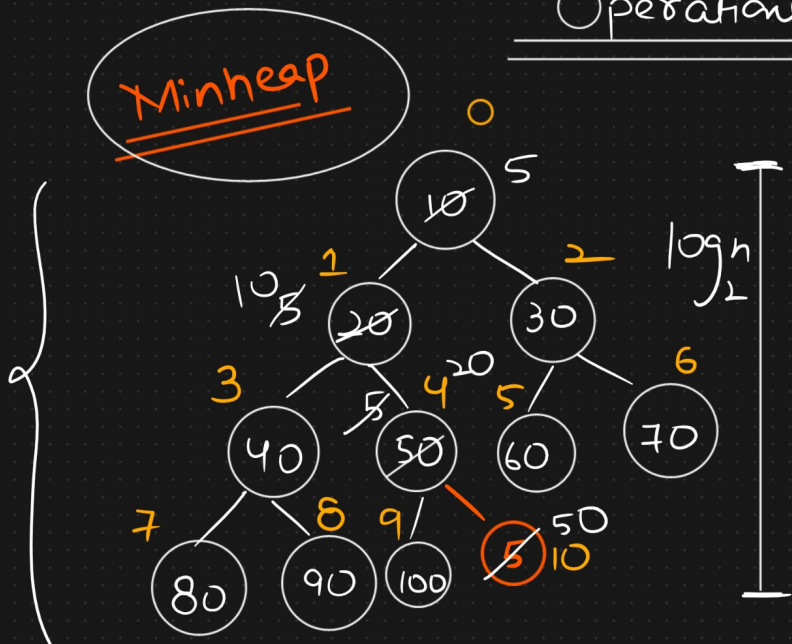


→ Valid Maxheap

Operations

1. Insertion

Minheap



$\log_2 n$ (new element = 5)

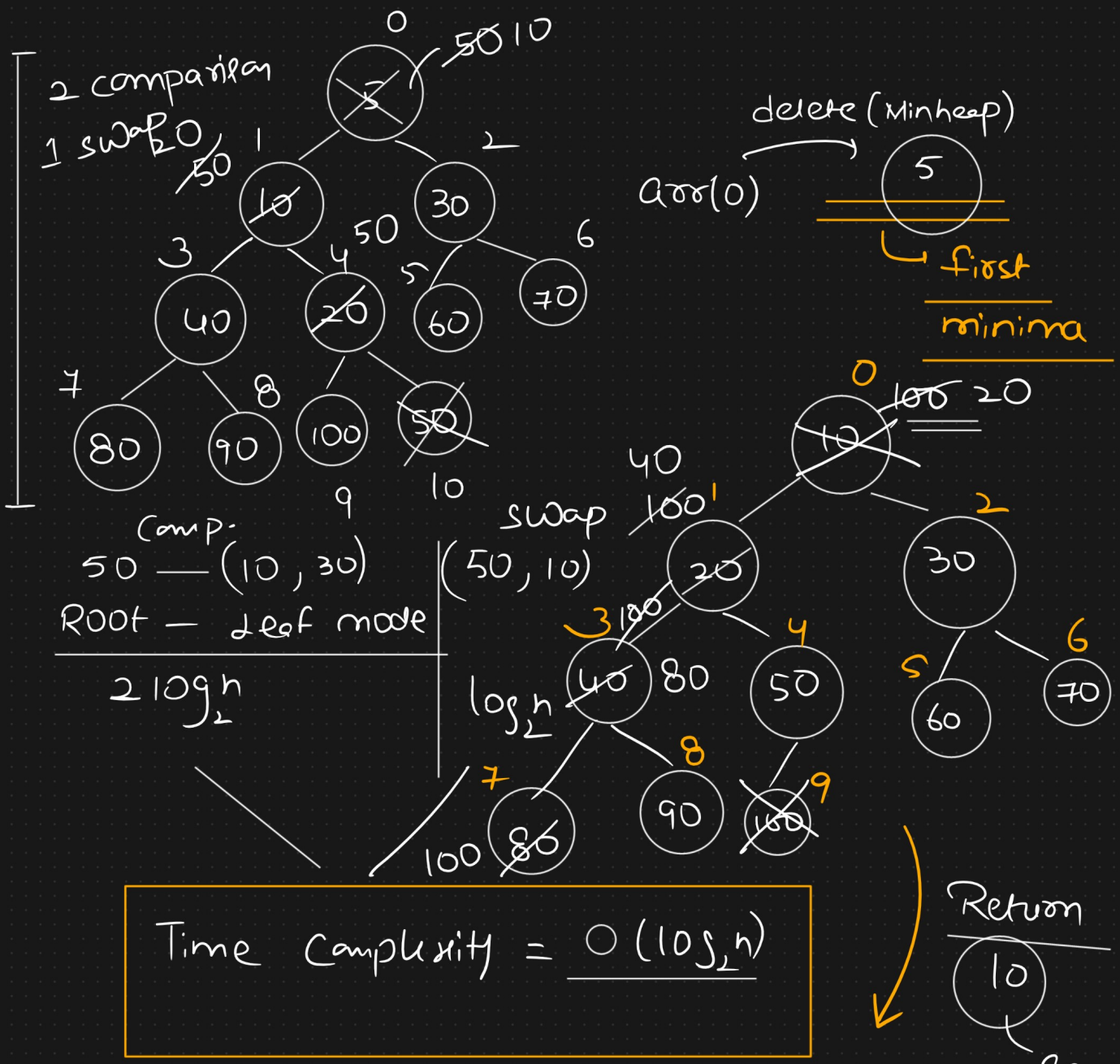
swap	comparison
1	50, 5
1	20, 5
1	10, 5

0	1	2	3	4	5	6	7	8	9	10
5 ~~10~~	10 ~~20~~	30	40	20 ~~50~~	60	70	80	90	100	50 ~~5~~

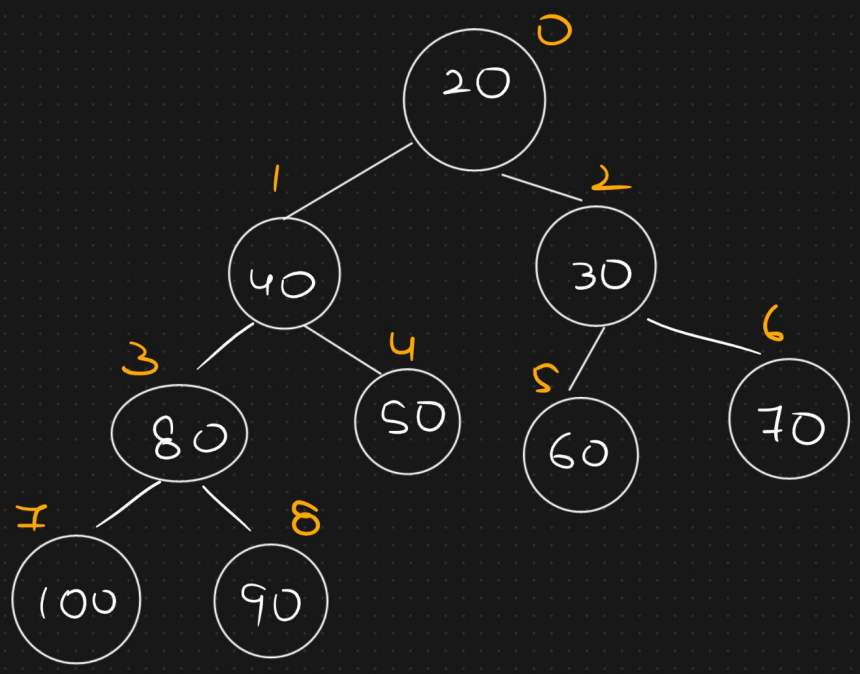
Time complexity = $O(\log_2 n)$

first minima ⟶ $O(1)$ ↳ Constant

Deletion



Time Complexity = $O(\log_2 n)$



HeapSort → Deletion

HeapSort → Sorting algorithm → Ascending order

Psuedocode — HeapSort

Time complexity

1. Create minheap → $O(n)$
2. Delete all the n-elements step-by-step
   - Store it in the RHS of the array
   - Print → Sorted array

$O(n \log_2 n)$

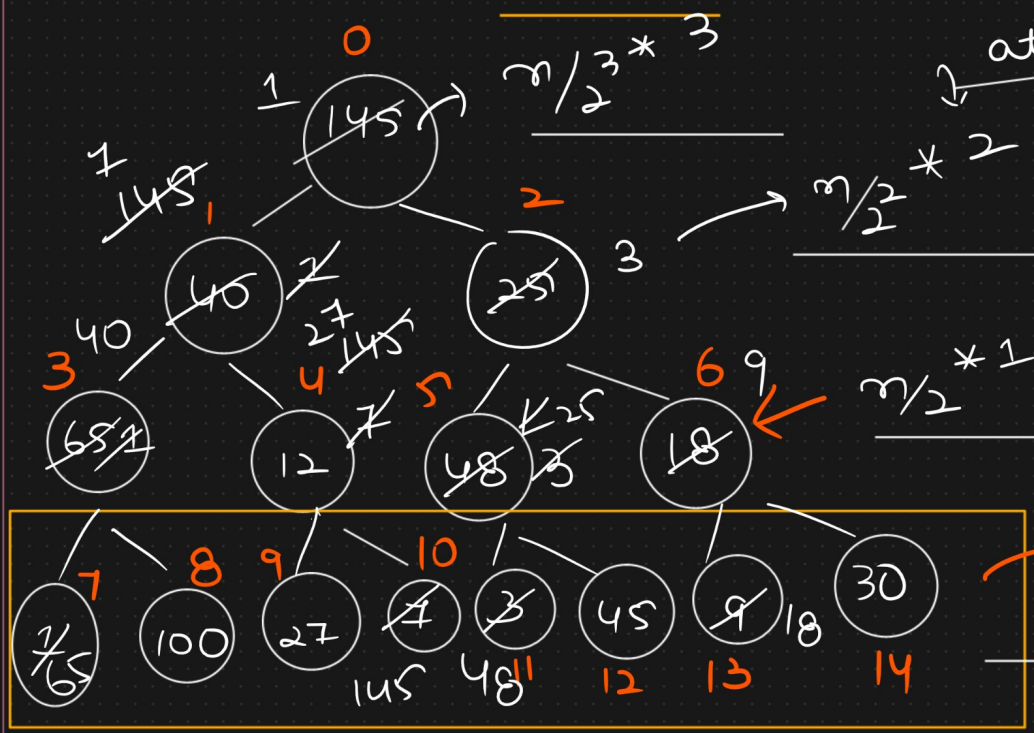
$$n + n \log n$$

$$\Rightarrow O(n \log n)$$

145, 40, 25, 65, 12, 48, 18, 1, 100, 27, 7, 3, 45, 9

30

Minheap



$\frac{n}{2^3} * 3$

at max

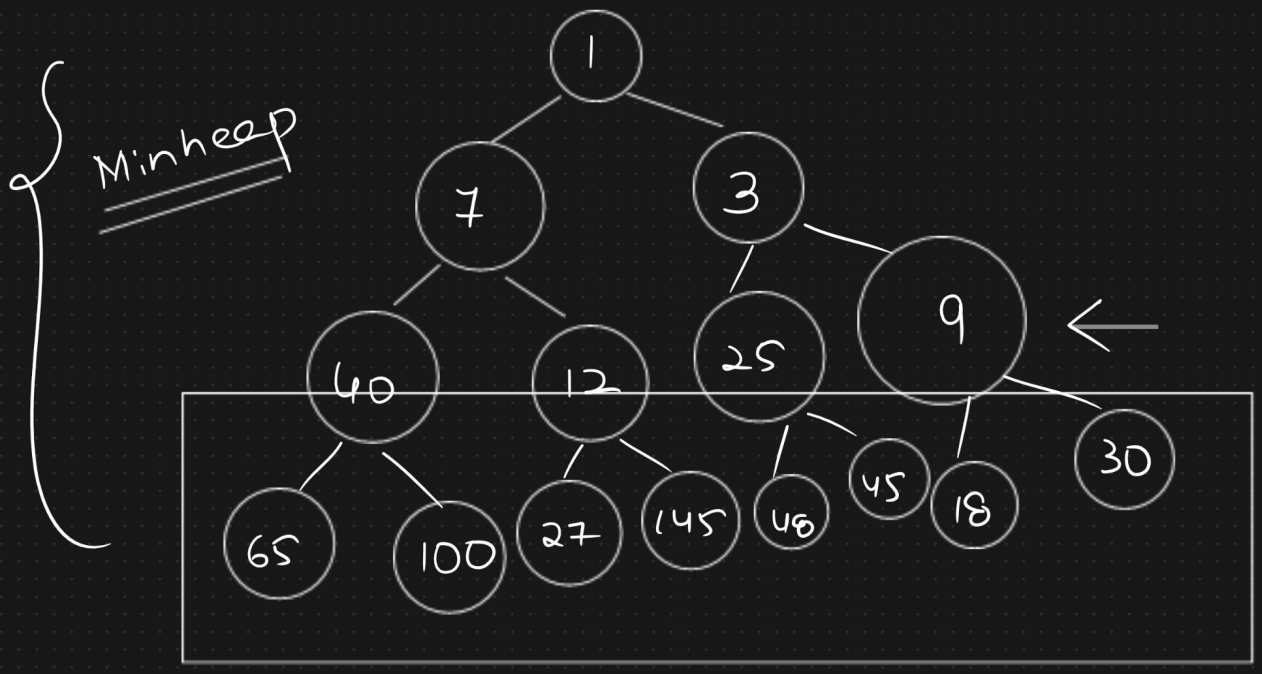
$\frac{n}{2^2} * 2$

$\frac{n}{2} * 1 = \#swaps$

0 swap

$\frac{n}{2^0} * 0$

Build Minheap → $O(n)$



Total Swaps $S =$

$$\frac{n}{2^0} * 0 + \frac{n}{2^1} * 1 + \frac{n}{2^2} * 2 + \dots + \frac{n}{2^{\log_2 n}} * \log_2 n$$

$$S = n\left(\frac{1}{2^1} + \frac{2}{2^2} + \frac{3}{2^3} + \dots + \frac{\log_2 n}{2^{\log_2 n}}\right) \quad \text{— (1)}$$

$$\frac{S}{2} = n\left(\frac{1}{2^2} + \frac{2}{2^3} + \frac{3}{2^4} + \dots + \frac{\log_2 n - 1}{2^{\log_2 n}} + \frac{\log_2 n}{2^{1+\log_2 n}}\right) \quad \text{— (2)}$$

(1) − (2)

$$\frac{S}{2} = n\left(\left(\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \cdots + \frac{1}{2^{\log_2 n}}\right) - \frac{\log_2 n}{2^{\log_2 n + 1}}\right)$$

$$r = \frac{1}{2} < 1$$

$$\text{Sum} = \frac{a(1 - r^n)}{1 - r}$$

$$n\left(\left(\frac{\frac{1}{2}\left(1 - \frac{1}{2^{\log_2 n}}\right)}{1 - \frac{1}{2}}\right) - \frac{\log_2 n}{2^{\log_2 n + 1}}\right)$$

$$2^{\log_2 n} = n^{\log_2 2} = n$$

$$\frac{S}{2} = n\left(\left(\frac{2^{\log_2 n} - 1}{2^{\log_2 n}}\right) - \frac{\log_2 n}{2^{\log_2 n} \cdot 2}\right)$$

$$\frac{S}{2} = n\left(\left(\frac{n-1}{n}\right) - \frac{\log_2 n}{n \cdot 2}\right)$$

$$\frac{S}{2} = \cancel{n} \cdot \frac{n-1}{\cancel{n}} - \cancel{n} \cdot \frac{\log_2 n}{\cancel{n} \cdot 2}$$

$$S = 2n - 2 - \log_2 n$$

$$\downarrow$$

$$O(n)$$

Recorded Videos

- Implementation ✓
- Gladden (12-2 PM) ✓
  - Interview Problems

Next week

- Tuesday → 7 to 9 PM
  - Interview Problems

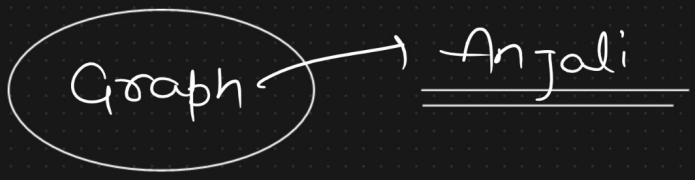
Priyabhatia@ineuron.ai

Logical Preepective

Implementation

Interview Quetion

- LinkedIn OR Email


Graph
Anjali