HEAPS

1) Heap is a complete Binary Tree
2) Head order property.

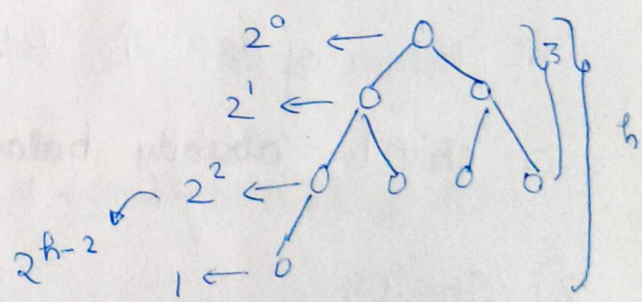
> **Note:**
> In a CBT, we can insert at ~~right~~ leftmost empty pos^n on last level. and we can delete the rightmost element in the last level only.

How CBT resolved the issues that we were facing with Balanced BST.

1) Height

Minimum no. of nodes with height h CBT →

let for h = 4

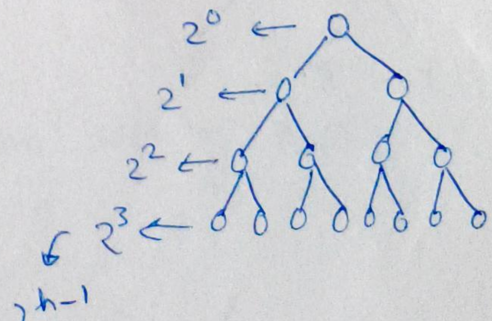


$$\underbrace{2^0 + 2^1 + --- + 2^{h-2}} + 1$$

$$\frac{2^0(2^{h-1}-1)}{2-1} + 1 = \boxed{2^{h-1}}$$

Maximum no. of nodes with height h CBT

let for h = 4



$$2^0 + 2^1 + --- + 2^{h-1}$$

$$= \frac{2^0(2^h - 1)}{2-1} = \boxed{2^h - 1}$$

Min nodes $\longrightarrow 2^{h-1}$

Max nodes $\longrightarrow 2^{h}-1$

$\therefore \quad \underbrace{2^{h-1} \le \overset{\text{Total number of nodes}}{n} \le 2^{h}-1}$

$2^{h-1} \le n$

$\log_2(2^{h-1}) \le \log_2 n$

$(h-1)\log_2(2) \le \log_2 n$

$h-1 \le \log_2 n$

$\boxed{h \le \log_2 n + 1}$

$\cancel{2^{h}-1 \le n}$

$n \le 2^{h}-1$

$n+1 \le 2^{h}$

$\log_2(n+1) \le \log_2(2^{h})$

$\boxed{\log_2(n+1) \le h}$

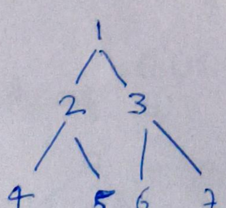
$\log_2(n+1) \le h \le \log_2 n + 1$

$O(\log_2 n) \le h \le O(\log_2 n)$

$\therefore$ Height of CBT is strictly $\log_2 n$.

$\therefore$ CBT is already balanced. $\rightarrow$ so no balancing ~~ovr~~ overhead.

2) Storing



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

parent index	child index
$0 \longrightarrow$	1, 2
$1 \longrightarrow$	3, 4
$2 \longrightarrow$	5, 6

$(2i+1, 2i+2)$

∴ we can store a CBT using an array.

→ we can also observe that,

if a parent node is at index i

→ then it's children are at $\underline{2i+1}$ & $\underline{2i+2}$.

$\uparrow$ left child $\uparrow$ right child.

Note:
we can maintain a nextIndex for insertion & deletion.

→ whenever we want to go from child to parent.

$$\frac{2i+1-1}{2} = i \quad , \quad \frac{2i+2-1}{2} = i$$

$$\boxed{\frac{\text{Child Index} - 1}{2}} \rightarrow \text{Parent Index}$$