CSE 101 3-3-22

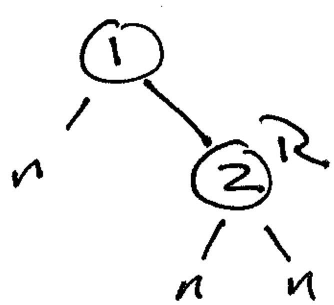
Pa8 : ext. 1 day

RBT insert:

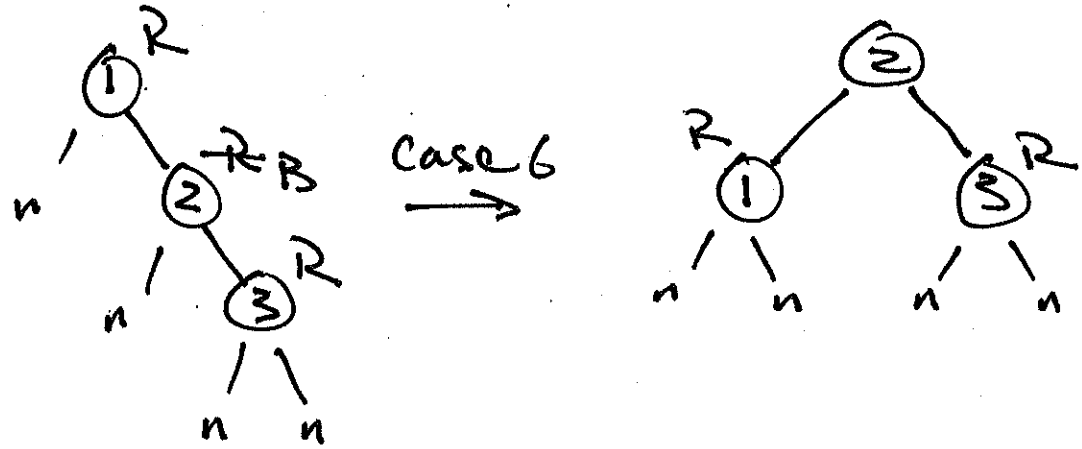
<u>Ex.</u> Keys 1 2 3 4 5 6

insert(1)

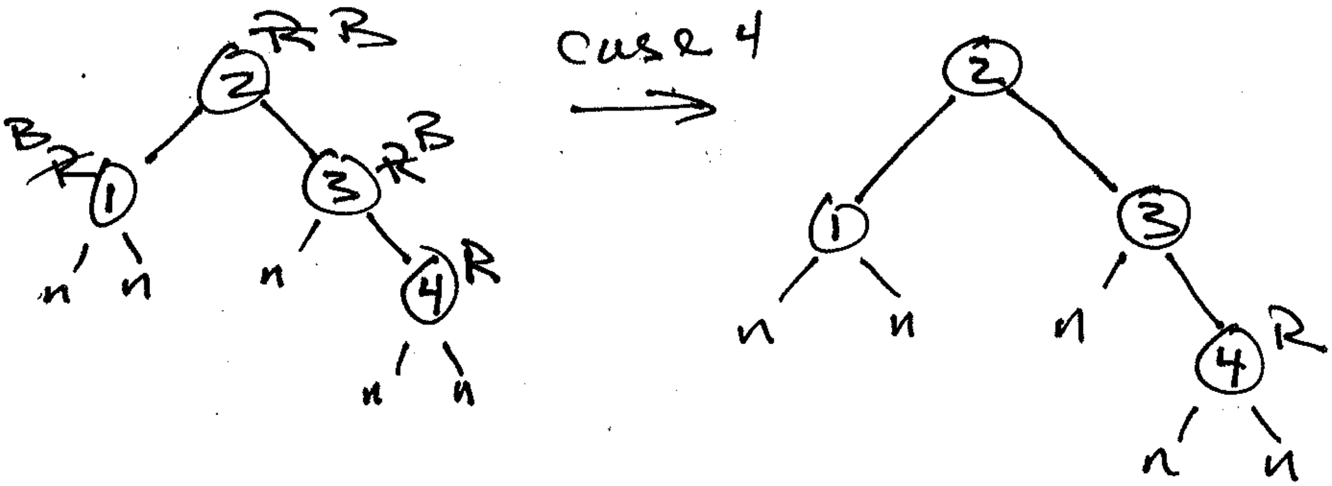

insert(2)



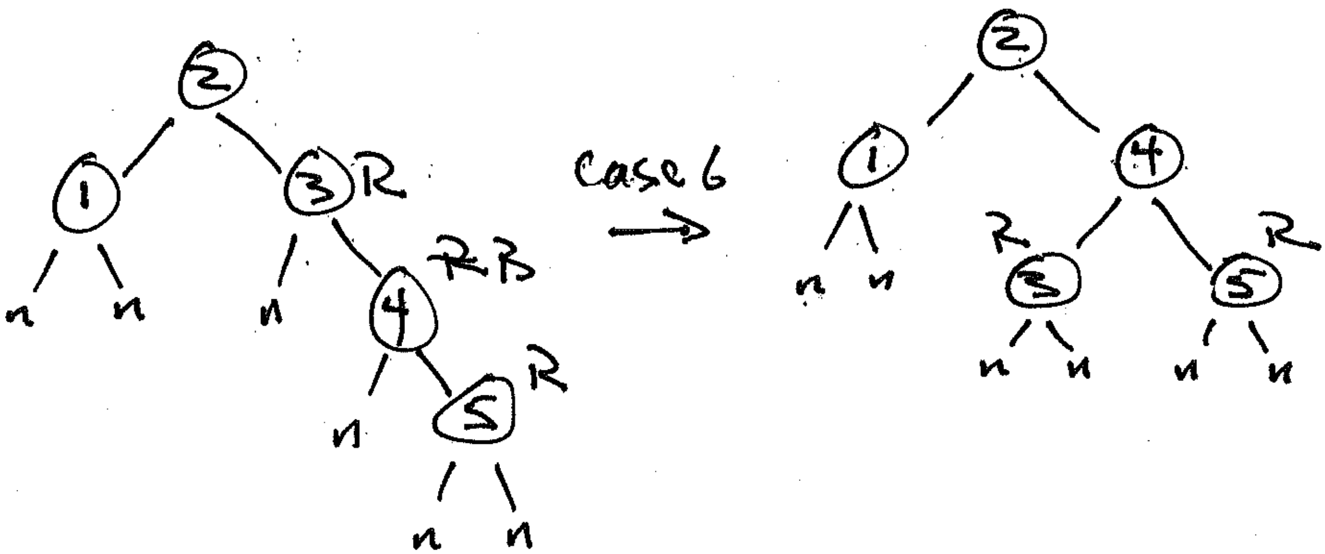
insert(3):



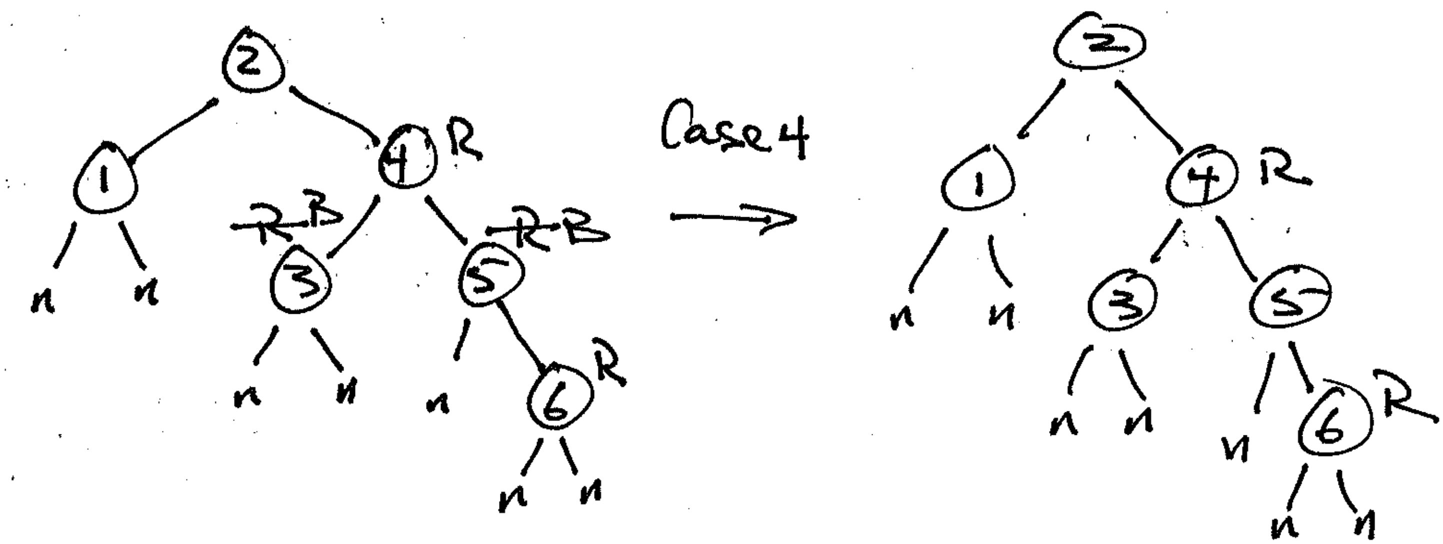

insert(4)



insert(5)




insert(6)

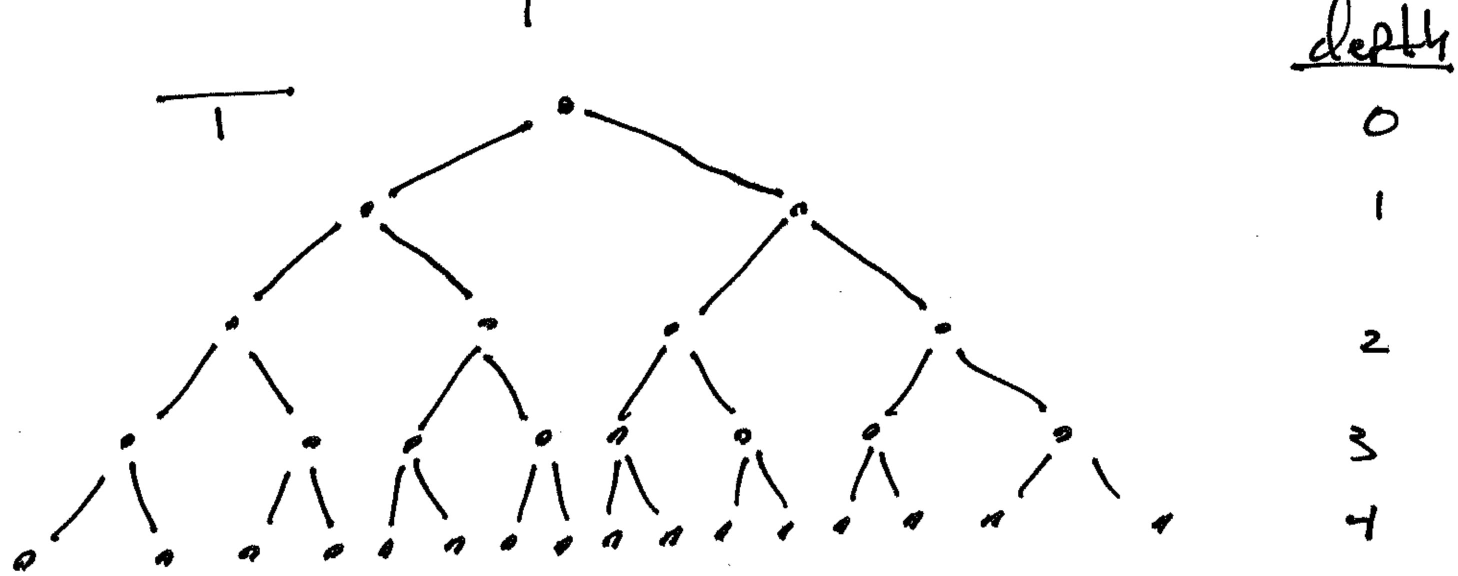


New data structure: <u>binary heap</u>

new ADT: <u>Priority Queue</u>

<u>more on binary trees:</u>

complete binary tree (CBT):



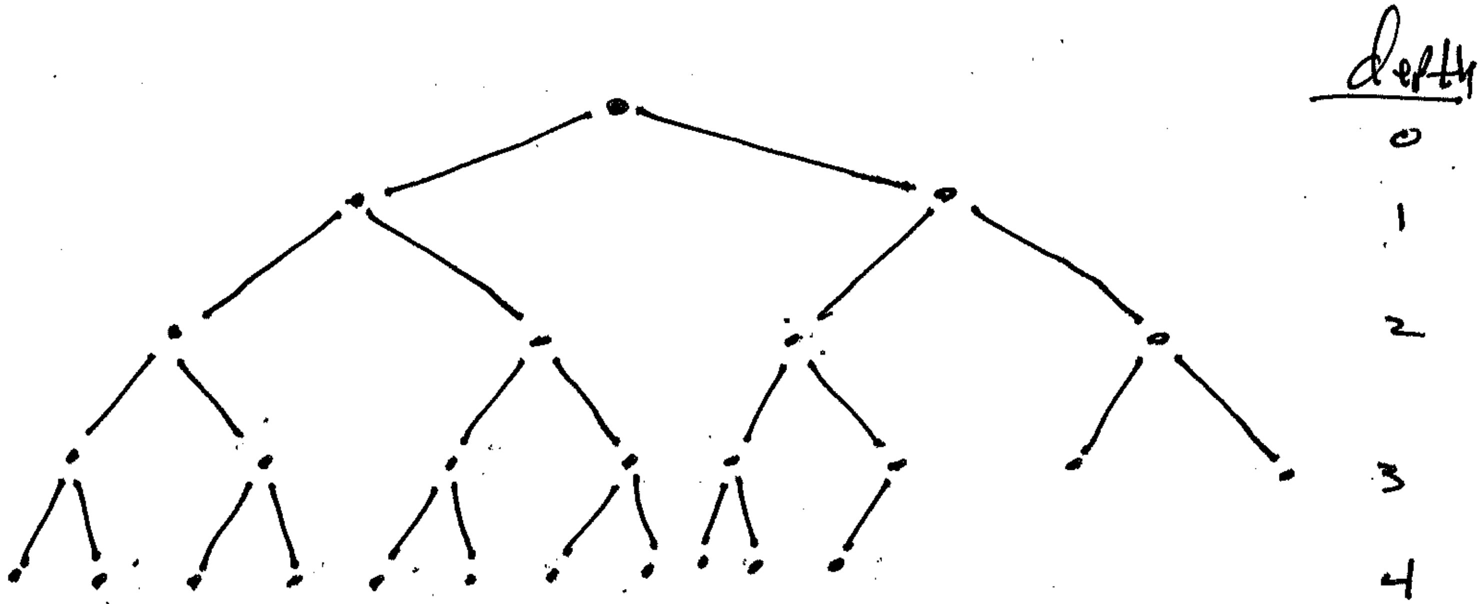

$(\#\text{ nodes at depth } d) = 2^d$. If T has height h, then

$$(\#\text{nodes}) = \sum_{d=0}^{d=h} 2^d = \frac{2^{h+1}-1}{2-1} = 2^{h+1}-1$$

Thus the height of a CBT on $n$ nodes is

$$h = \lg(n+1) - 1$$

almost complete binary tree (ACBT)




The # of nodes $n$ in an ACBT of height $h$ must satisfy

$$2^h - 1 < n \leq 2^{h+1} - 1$$

$\therefore \quad 2^h \le n < 2^{h+1}$

$\therefore \quad h \le \lg(n) < h+1$

$\therefore \quad h = \lfloor \lg n \rfloor$

Recall Theorem

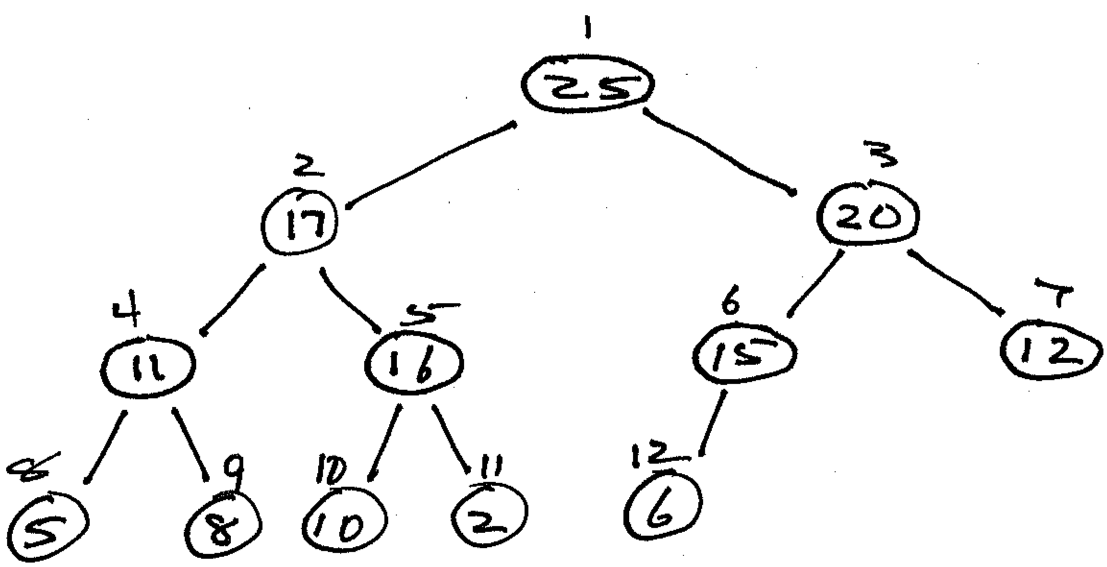
Any Binary Tree with $n$ nodes and height $h$ satisfies

$$h \ge \lfloor \lg n \rfloor$$

## 6.1 Heaps

A Binary Heap is an array object which represents an ACBT. each node in tree corresponds to an array element. (some array elements may not correspond to nodes.

EX



A

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 17 | 20 | 11 | 16 | 15 | 12 | 5 | 8 | 10 | 2 | 6 | 0 | 0 | 0 | 0 |

Array attributes:

- length[A]
- heapsize[A]

helper functions:

$$\text{Parent}(i) = \left\lfloor \frac{i}{2} \right\rfloor \qquad (i \geq 2)$$

$$\text{left}(i) = 2i$$

$$\text{right}(i) = 2i + 1$$

Array values satisfy:

- max-heap property: $A[\text{Parent}(i)] \geq A[i]$

<u>or</u>

- min-heap property: $A[\text{Parent}(i)] \leq A[i]$

we assume our heaps are all max heaps.

Heap operations

- Heapify : maintain the heap property
- BuildHeap : create a heap from an un-ordered array
- Heapsort : sort an array
- Priority Queue ops :

  ⋮

  next time . .