CSE 101 2-23-21

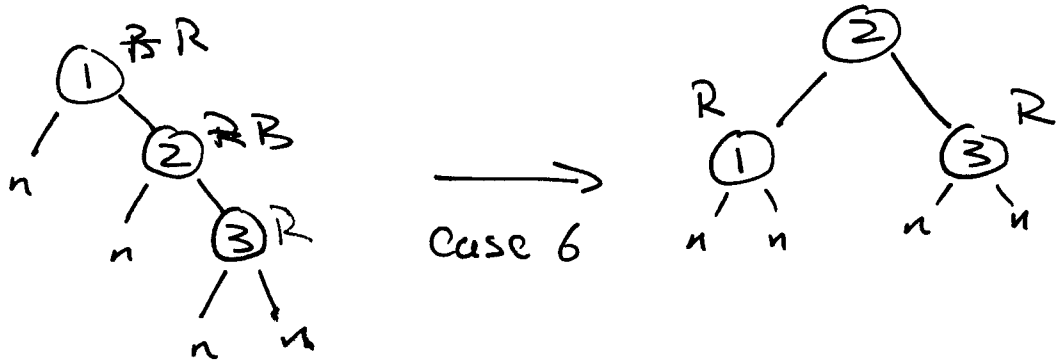
Ex RBT insertion

Keys: 1, 2, 3, 4, 5

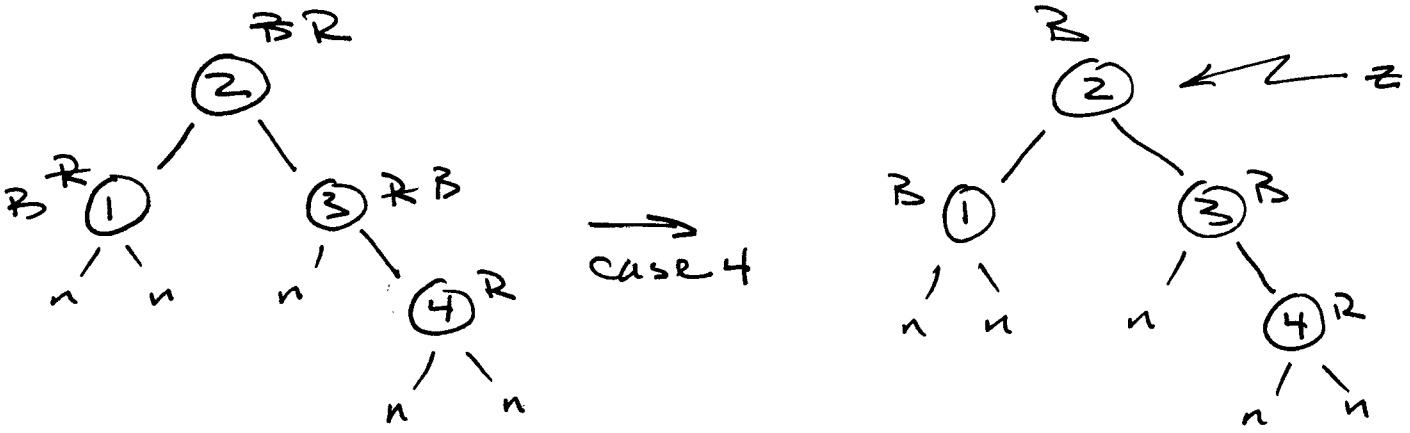
insert 1

insert 2

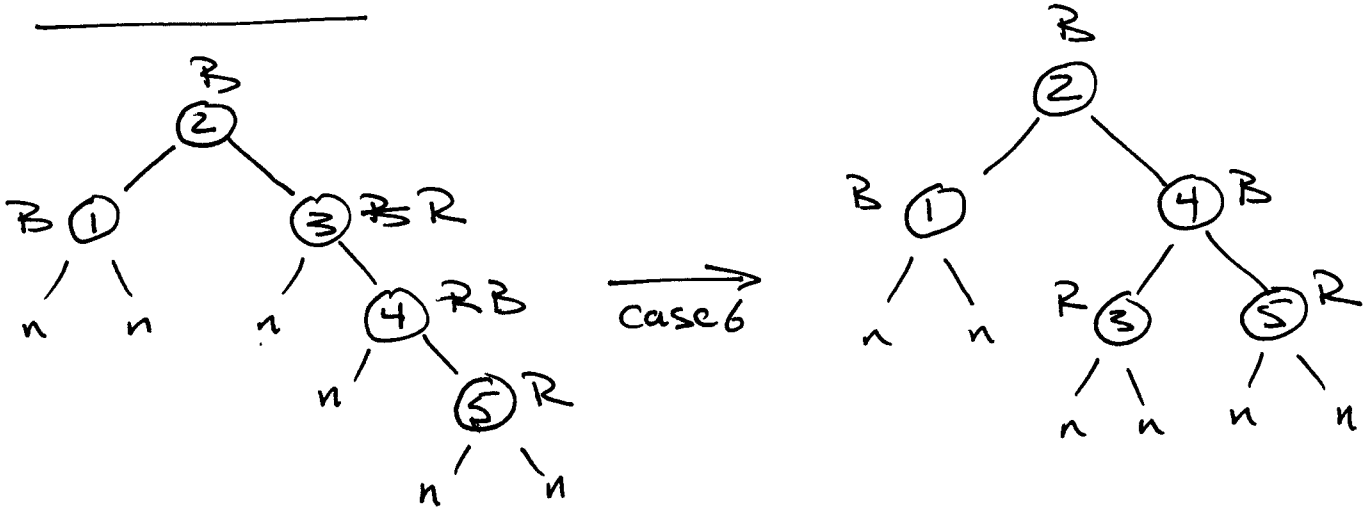
insert 3



insert 4



insert 5



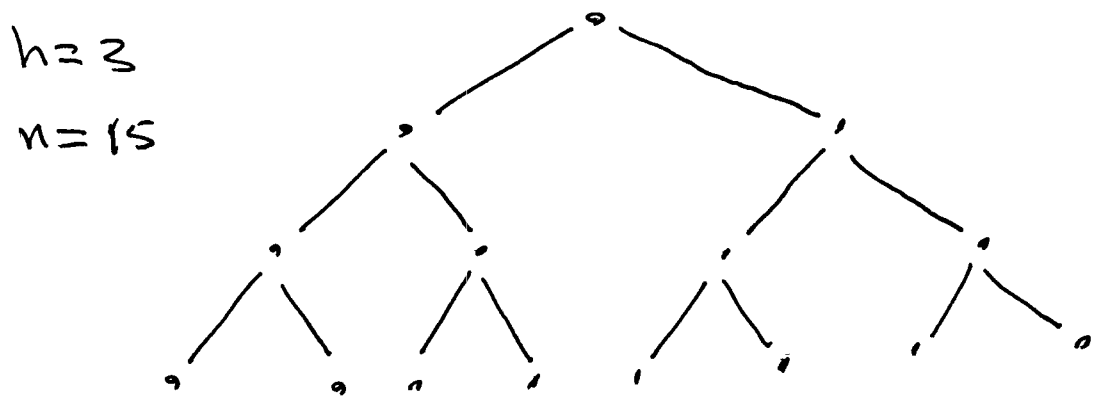
Chap. 6 : Heaps, Heapsort, Priority Queues.

Data struct — Heaps

ADT — Priority Queues

Recall

complete binary tree (CBT)



depth	# nodes
0	1
1	2
2	4
3	8

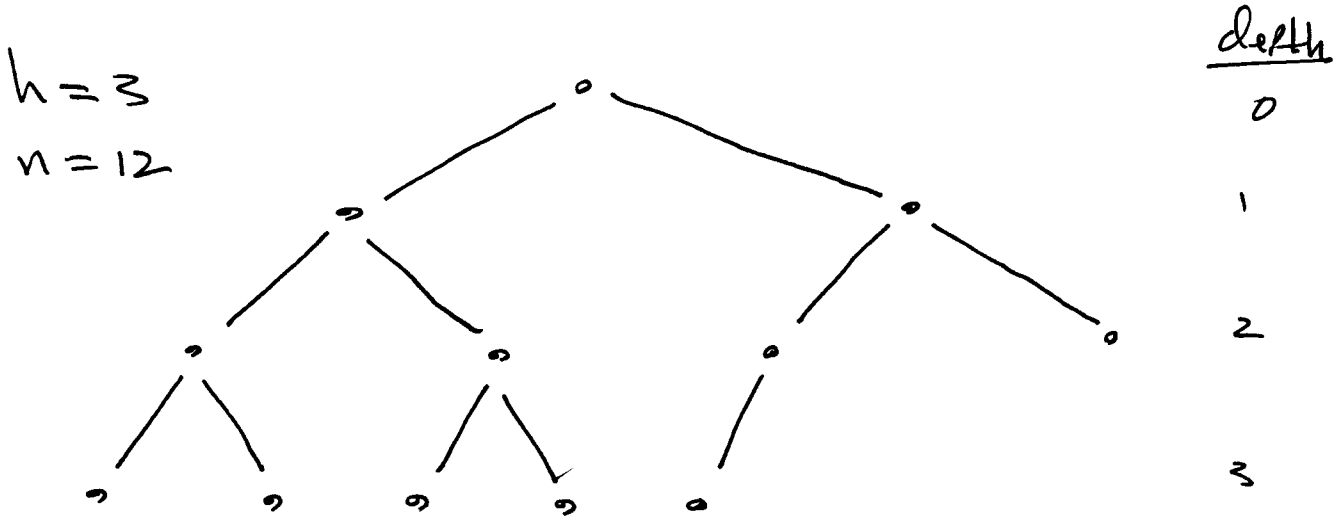
- all internal nodes have 2 children
- all leaves at same depth.

(# nodes at depth i) $= 2^i$

$$n = \sum_{d=0}^{h} 2^i = \frac{2^{h+1}-1}{2-1} = 2^{h+1} - 1$$

almost complete binary tree (ACBT)

- all levels are filled, except (possibly) the last.
- deepest level is filled left to right



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

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

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

$$h \leq \lg(n) < h+1$$

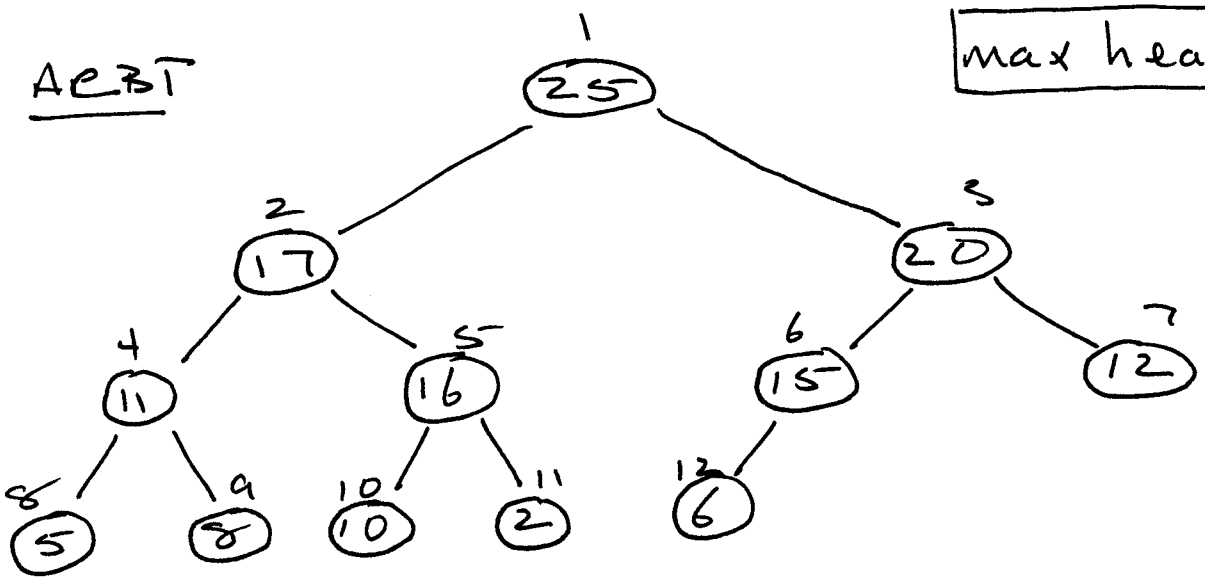
$$\therefore \quad h = \lfloor \lg(n) \rfloor$$

(6.1) Heaps

A Binary Heap is an array that represents an ACBT. each node corresponds to an array element. (some array elements might not correspond to nodes.)

Ex. ACBT

max heap



Array

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	25	17	20	11	16	15	12	5	8	10	2	6	•	•	•	•

length[A] = 16

heapSize[A] = 12

Array Attributes:

- length[A]
- heapSize[A]

Helper functions:

- $Parent(i) = \lfloor \frac{i}{2} \rfloor$ $\quad (i \geq 2)$
- $left(i) = 2i$
- $right(i) = 2i + 1$

Heap Properties:

- max-heap Property: for $1 \leq i \leq heapSize$

$$A[Parent(i)] \geq A[i]$$

or

- min-heap Property: for $1 \leq i \leq heapSize$

$$A[Parent(i)] \leq A[i]$$

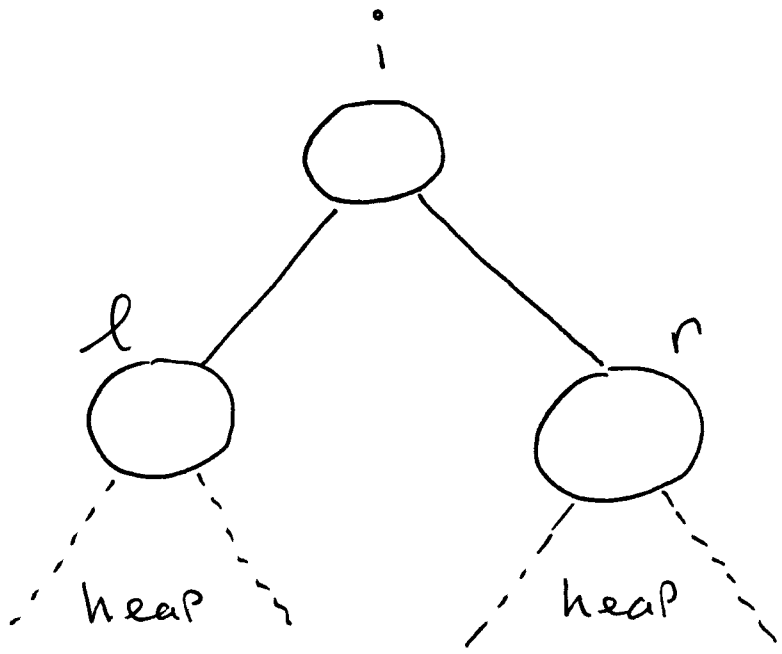
Algorithms:

- Heapify
- BuildHeap
- HeapSort
- P.Q. algorithms:
  - Insert
  - Max
  - ExtractMax
  - IncreaseKey

16.2) Heapify

Goal of Heapify(A, i) cause max heap prop. to be satisfied at index i (and in left & right sub-trees of i).

Precond: have valid heaps at subtrees left(i) and right(i).



See pseudocode on webpage.

Let $T(n)$ = worst case runtime of heapify when run on a subtree (rooted at i) with n nodes. ≥ 0

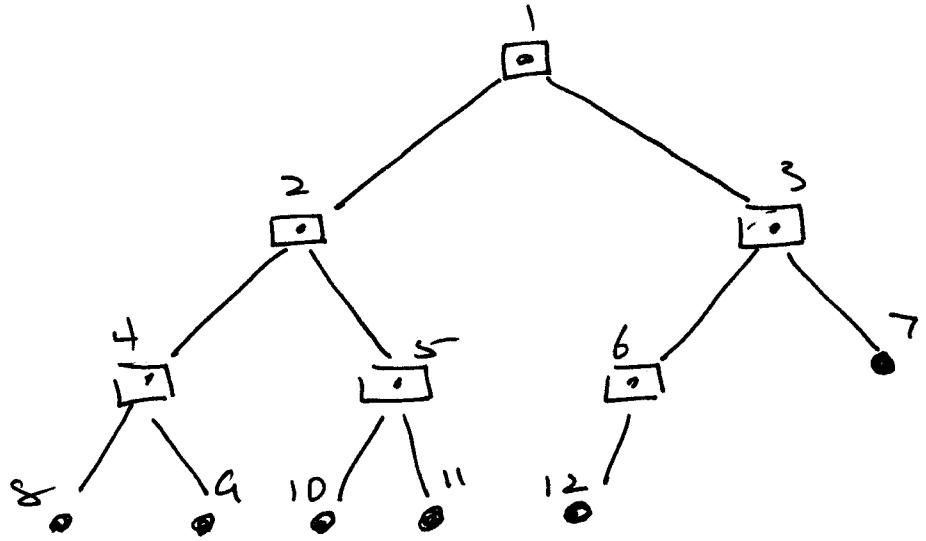
$T(n) = (\text{height of sub-tree at } i) = \lfloor \lg n \rfloor$

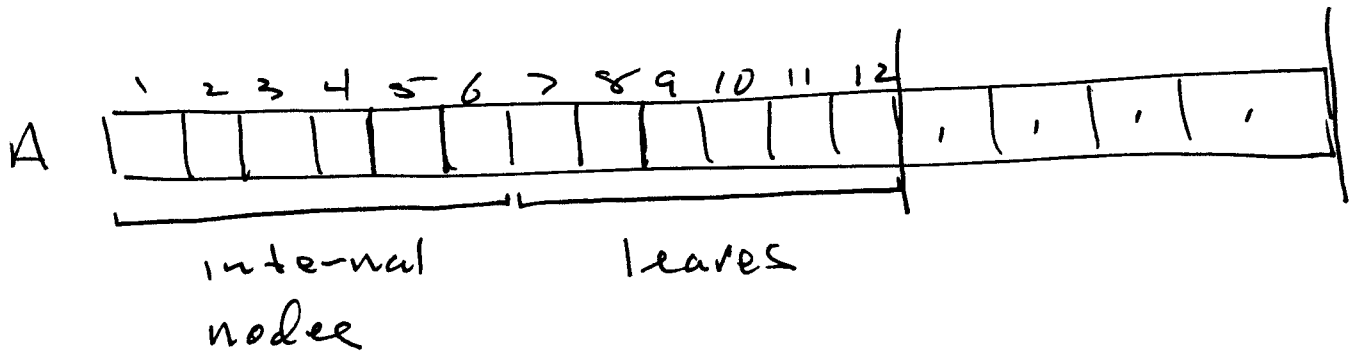
$= \Theta(\log(n))$

$= \Theta(h)$

(6.3) BuildHeap

Ex





see pseudo-code.

Exercise: start with an un-ordered array, and run BuildHeap( ).