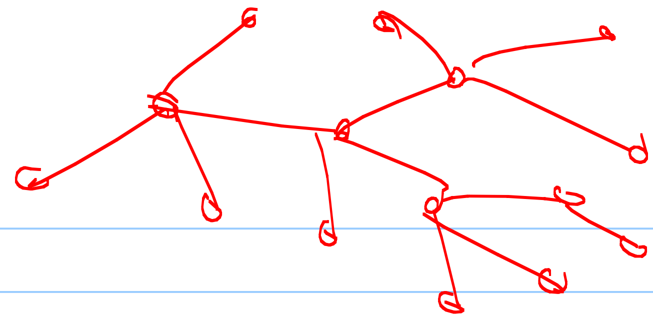
CS 180 - Tree stuff

Note Title 11/4/2011

Announcements

- Program 5 due tomorrow
- HW3 up today/tomorrow

Last time: Trees



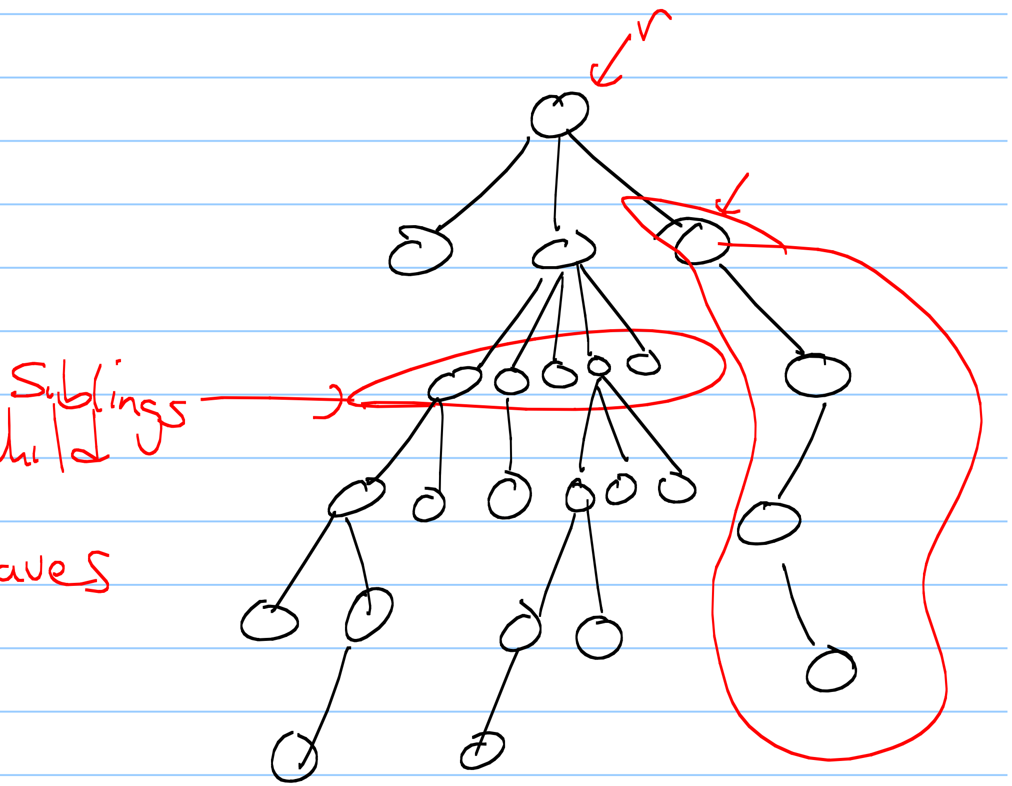
Dfn: A tree T is a set of nodes storing elements in a parent-child relationship.

T has a special node r, called the root.

Each node (except r) has a unique parent.

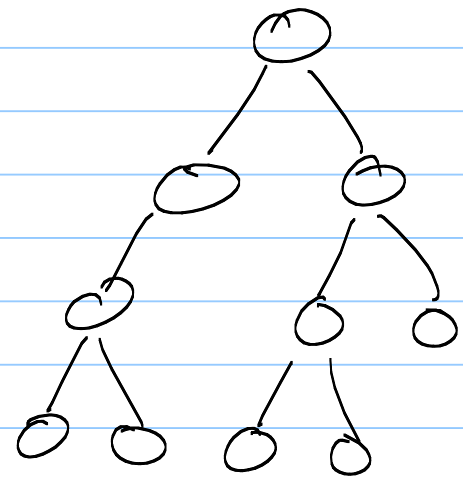
More dfns

- child
- siblings : share common parent
- leaves : vertices w/ no child
- internal nodes : not leaves
- rooted subtree
- desendant / ancestor



Binary Tree

- Every node has ≤ 2 children.



- left
- right

} pointers

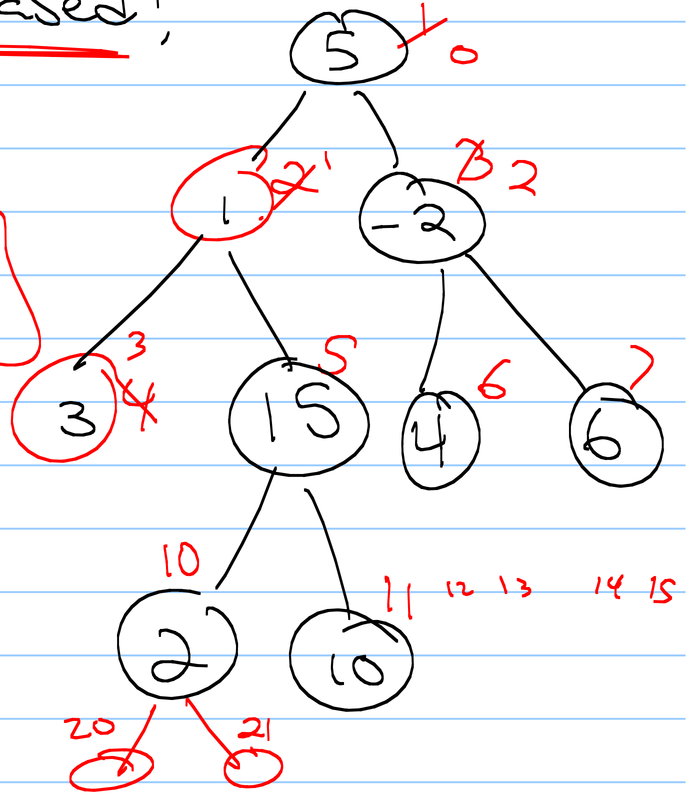
code for this will be written next week

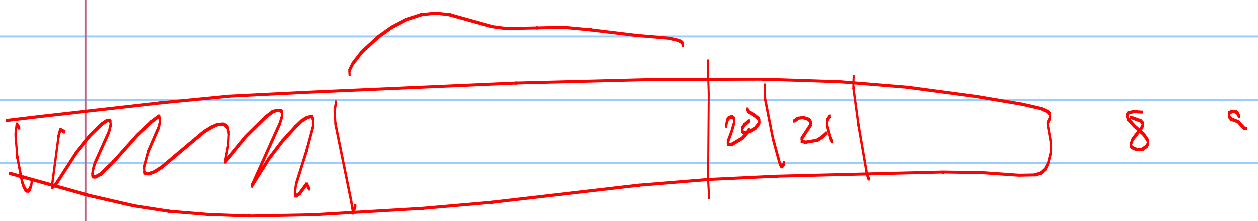
Nice trick

Can be pointers or array based!

left(x) = 2x+1
right(x) = 2x+1

5	1	-2	3	15	4	6			2	10
0	1	2	3	4	5	6	7	8	9	10





Depth & Height - defined recursively

depth: depth(r) = 0

depth(v) = depth(parent(v)) + 1

depth(tree) = max depth

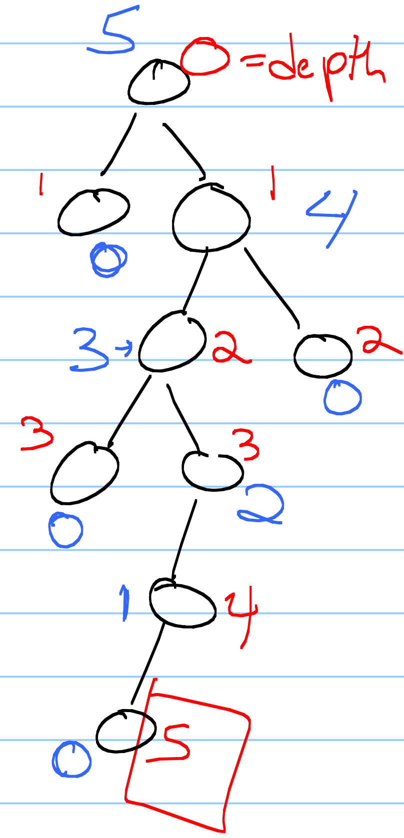
height: height(leaf) = 0

height(v) = max(height of children) + 1

height(tree) = height(r)

height(T) = depth(T)

height(v) ≠ depth(v)



Data Structures (for trees)

Priority Queue: supports the following operations

(push) insert(e) : adds element e to the data structure

$O(n)$

(pop) removeMax() : removes maximum element

(top) getMax() : returns maximum element

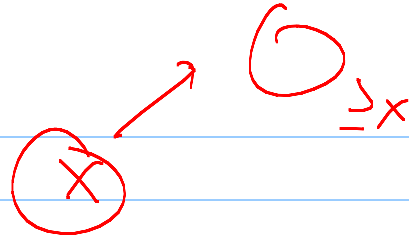
(size, empty)

How to build?

use lists or vectors & sort

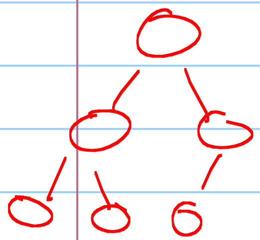
(something will take $O(n)$ time)

Heaps



A binary tree where:

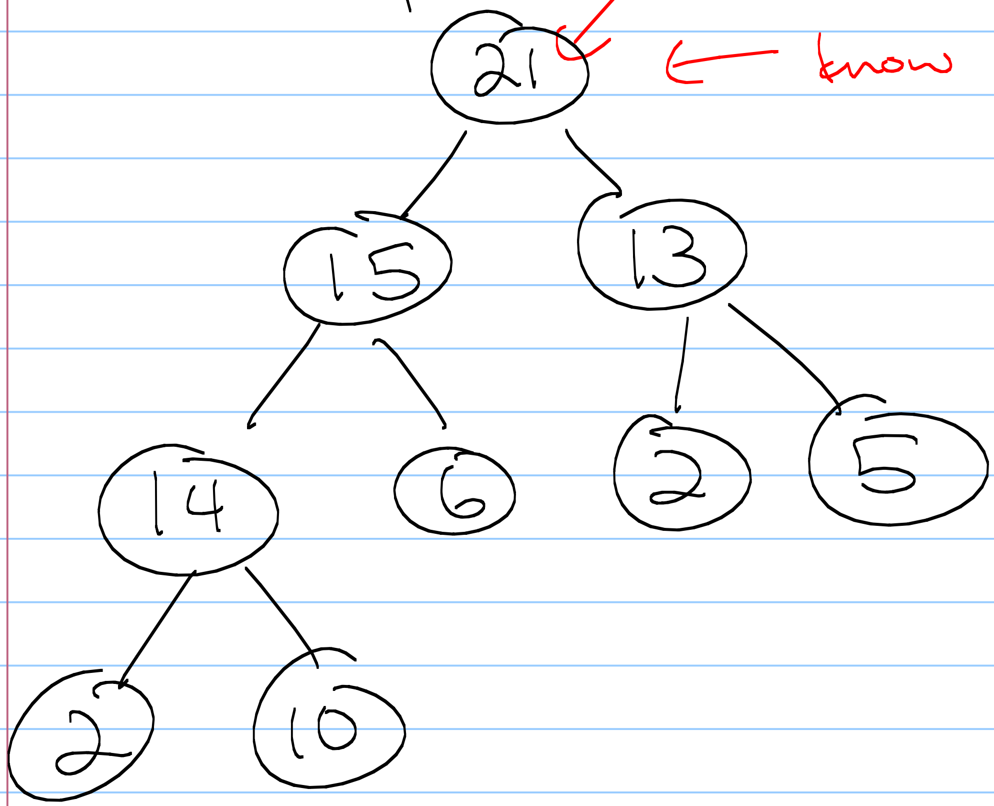
- For every node v (other than root), the key stored at v is $\leq$ key stored at v's parent



- The tree is complete: levels 0 to $h-1$ are full, and level h is filled in left to right order

Max Heap

return this for getMax



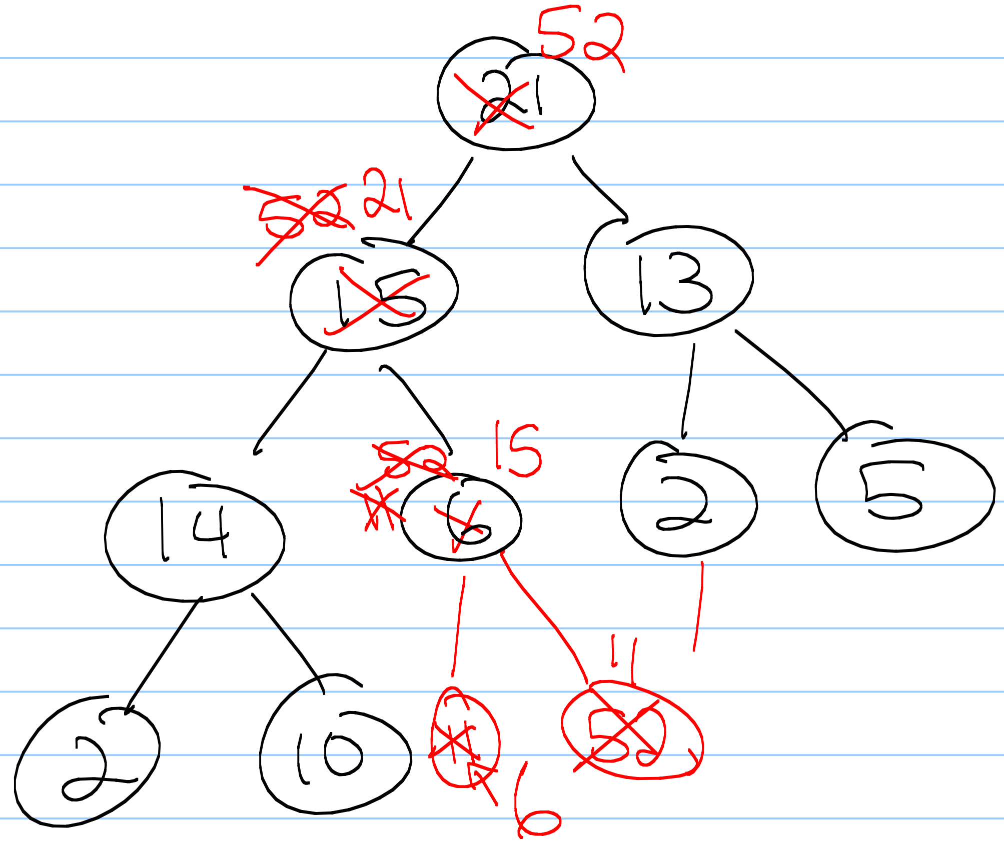
← know is ≥ children

Insert

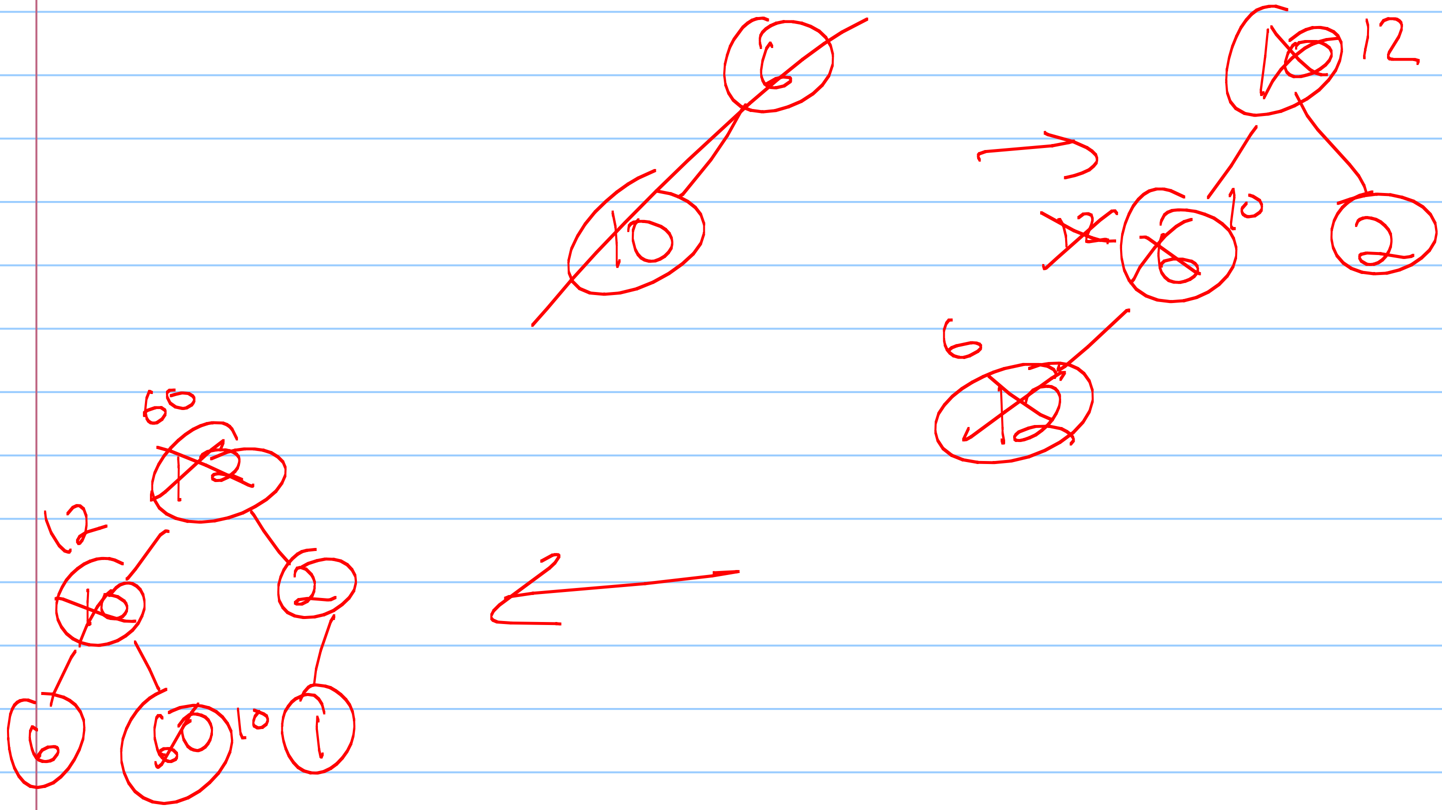
insert (11)

insert (52)

" bubbling up "

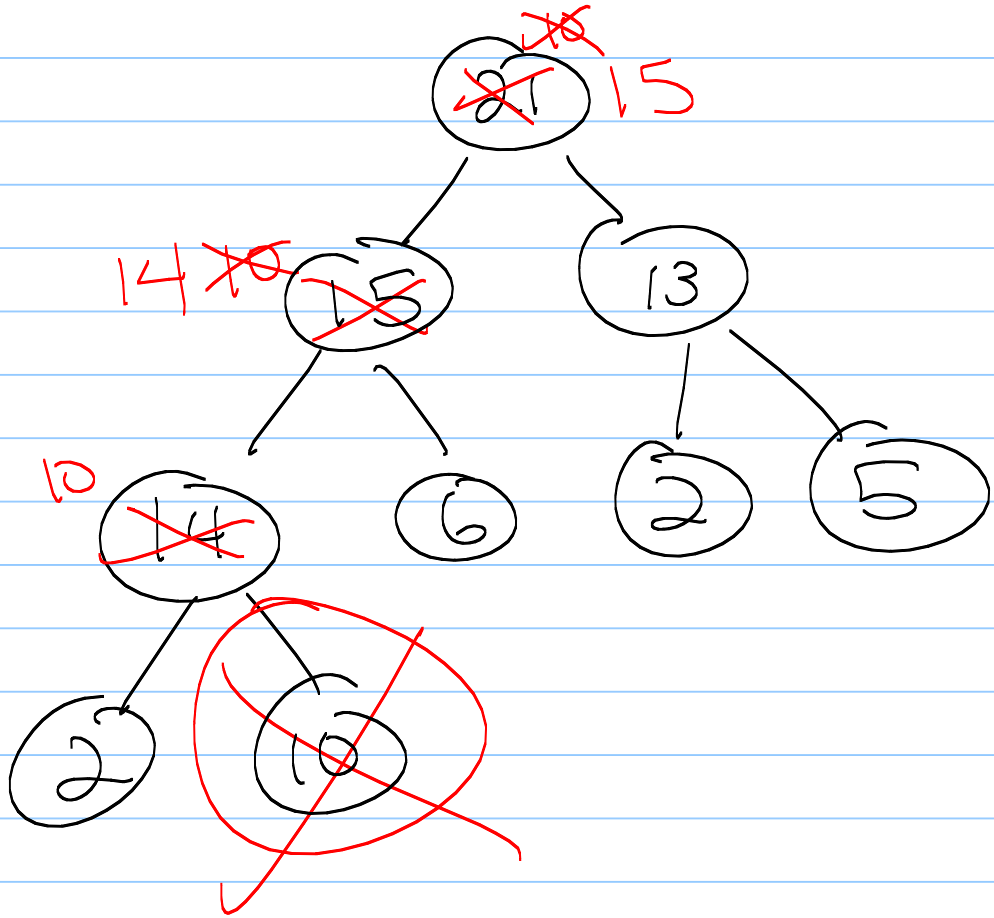


Insert : ~~6~~, ~~10~~, ~~2~~, ~~12~~, ~~60~~, 1



Remove Max

"bubble down"



Running times

How many comparisons/swaps?

$O(h)$ (or $O(d)$)

$h = \lceil \log_2 n \rceil$

total $O(\log_2 n)$ for any operation

versus $O(1)$ or $O(n)$

Binary Search Trees