

CS180 - Binary Search Trees

Note Title

10/25/2013

Announcements

- HW is up

- Turn in exam redo

Last time: Priority Queues

- insert(e): add e to our data structure
- get Max(): return element with maximum key (its e)
- remove Max(): delete element with maximum key

With vectors or lists: $O(n)$ (for one function)

Last time: Heaps

A binary tree where we maintain
2 invariants:

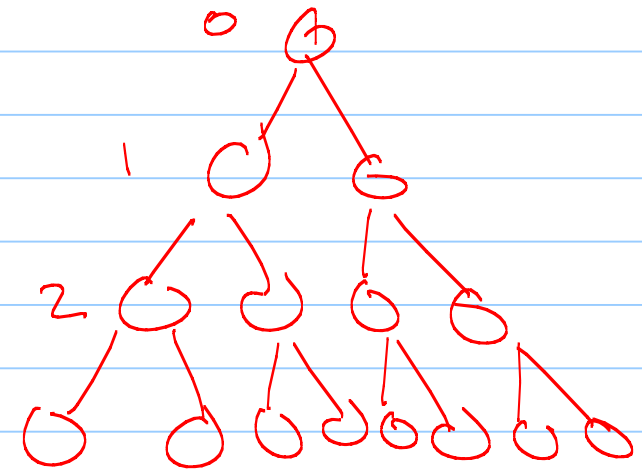
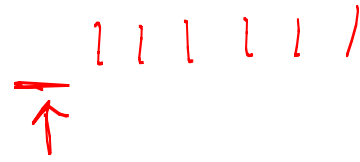
- Tree is complete.
- Any node's value is \leq its parent's value.

Runtimes: $O(\log n)$ for insert & delete
 $O(1)$ - get Max

(Code is on webpage.)

Complete trees:

each level has 2^i nodes

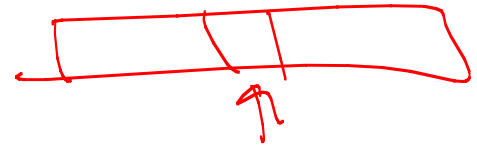


$$n = \sum_{i=0}^{\text{depth } d} 2^i = 2^{d+1} - 1 \quad d \rightarrow 3$$

depth?

$$\begin{aligned} n &= 2^{d+1} - 1 \\ n+1 &= 2^{d+1} \\ \Rightarrow \log_2(n+1) &= d+1 \approx d = \lceil \lg_2 n \rceil \end{aligned}$$

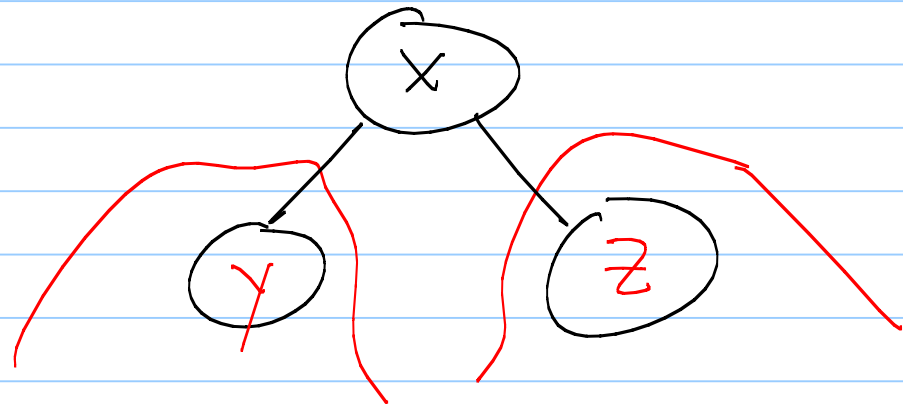
Binary Search Trees



A binary tree where we maintain the following:

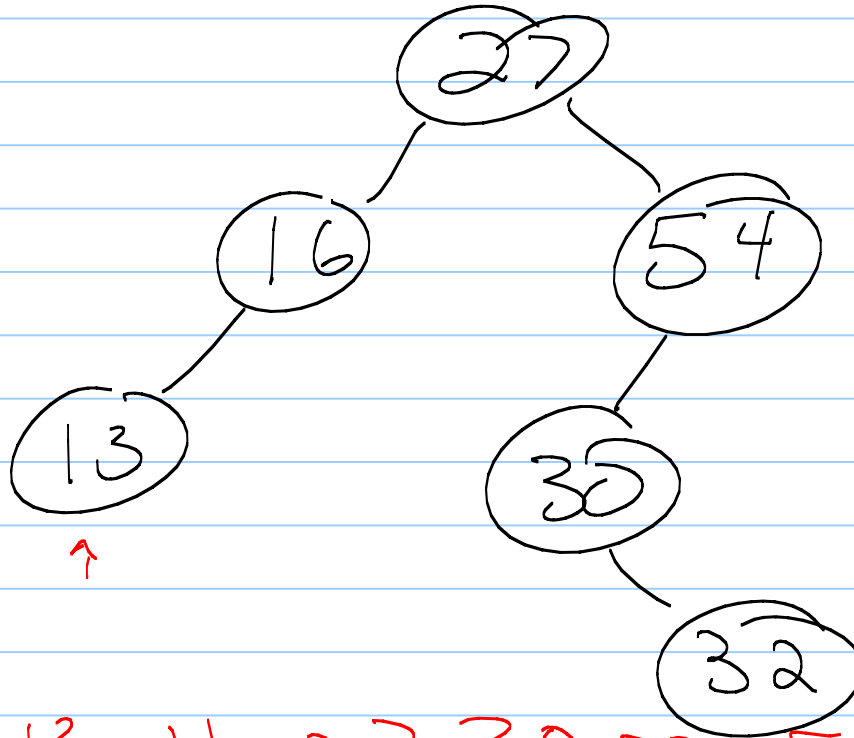
The value at any node is \geq its left child and $<$ its right child.

$$y \leq x$$
$$z > x$$



Example :

Sorted order: in order



go left
print myself
go right

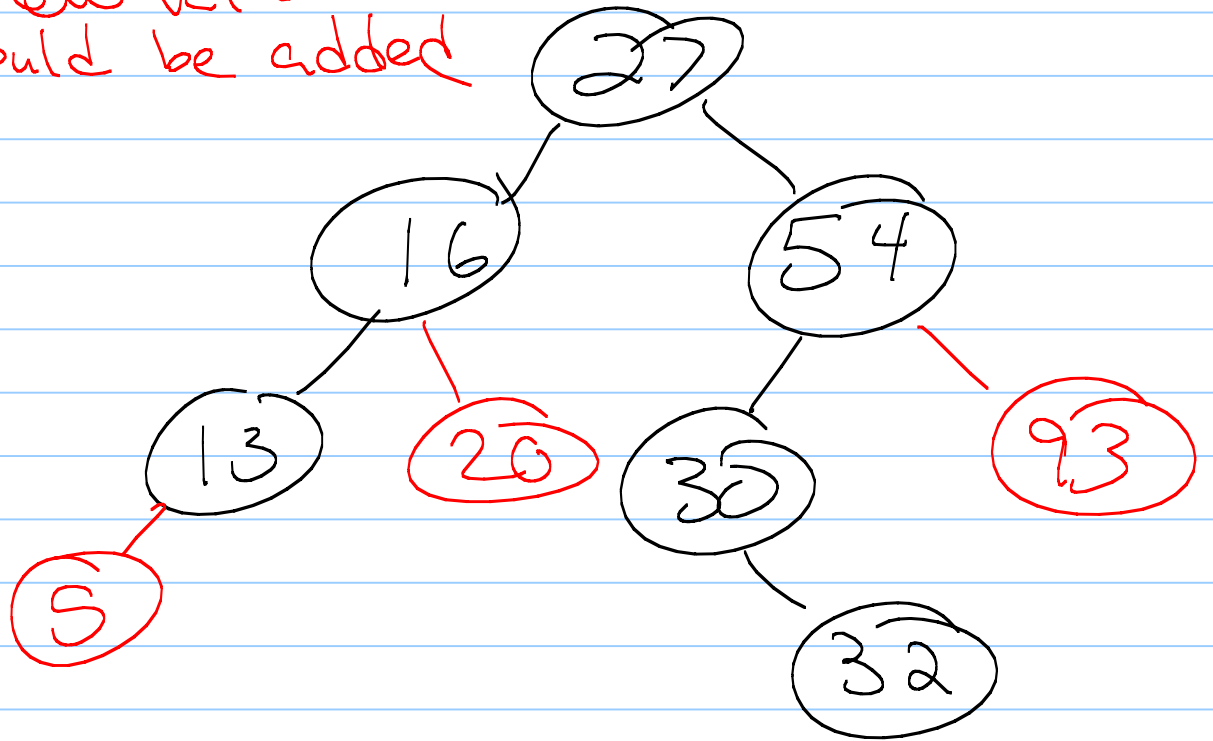
13, 16, 27, 30, 32, 54

Insert: unique location in current tree where new value could be added

insert (20)

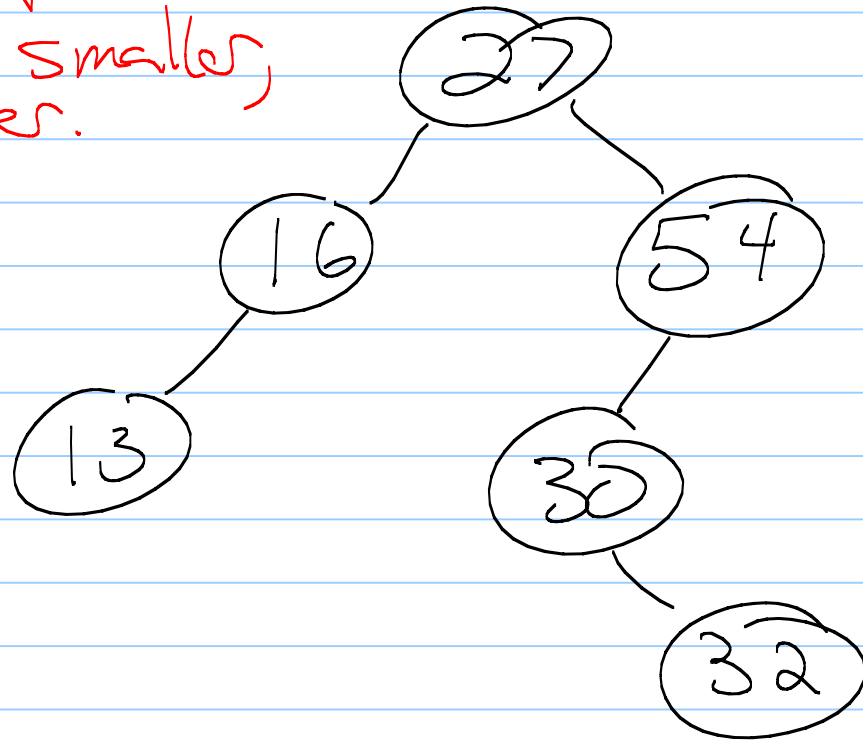
insert (5)

insert (93)



Find : traverse tree,
comparing &
moving left if smaller,
right if greater.

find(62)



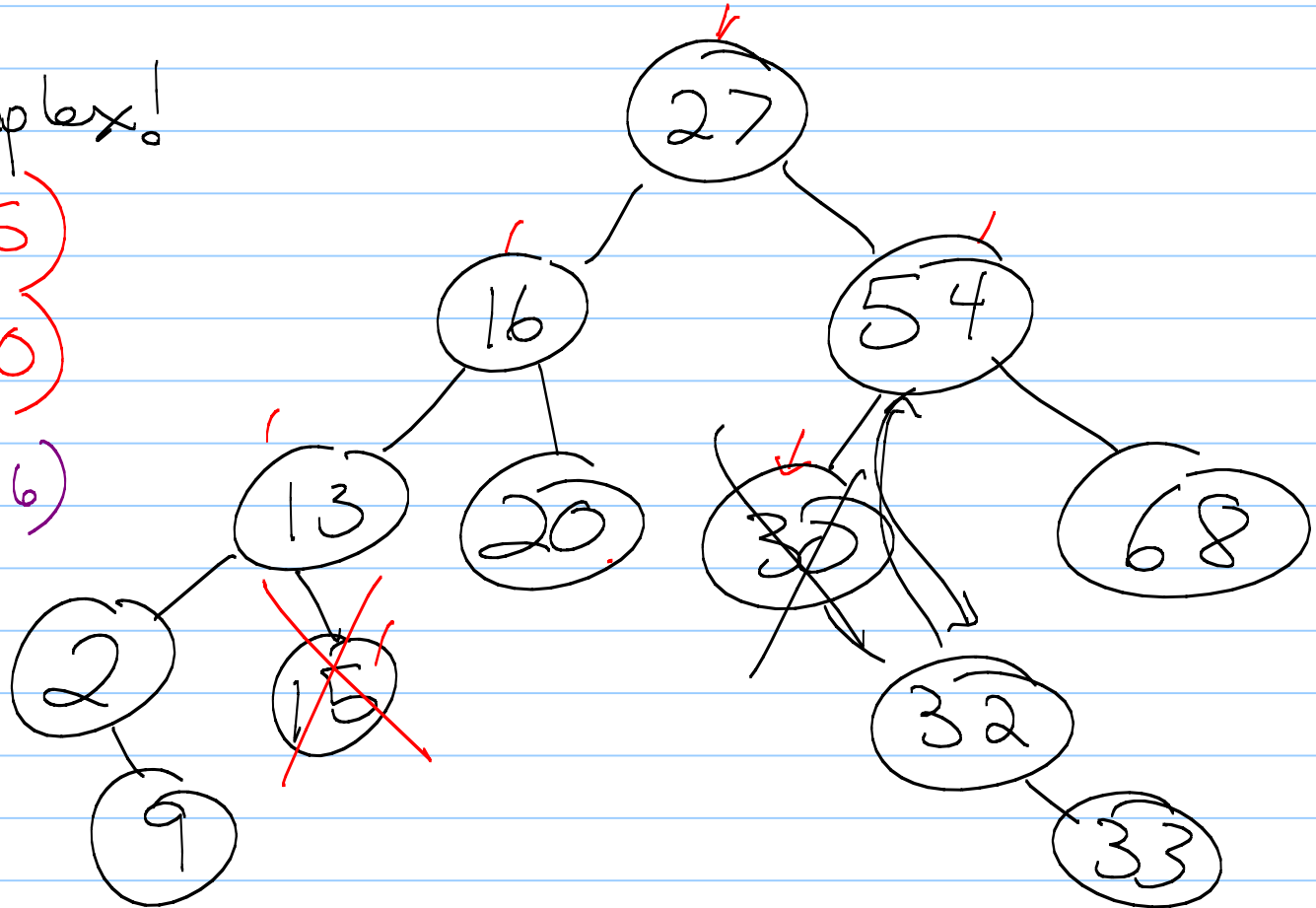
Delete:

More complex!

delete (15)

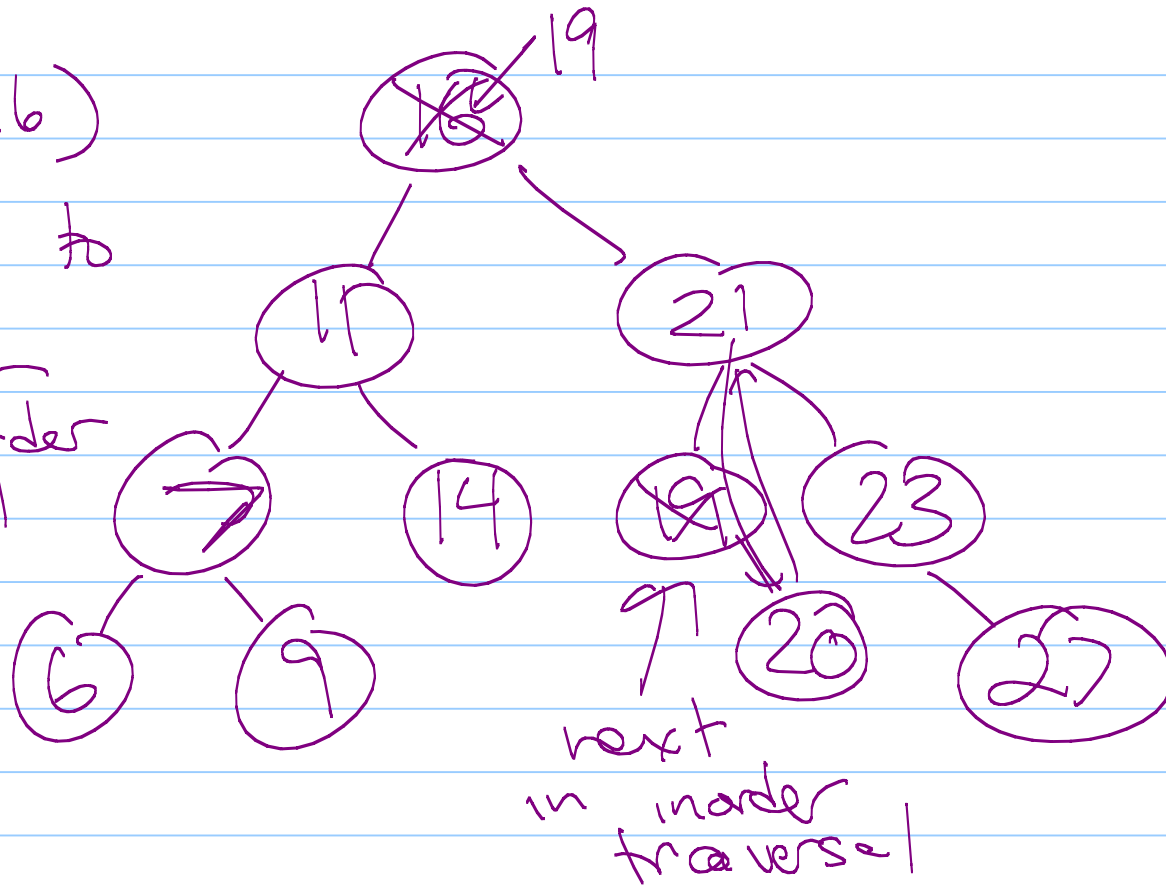
delete (30)

delete (16)

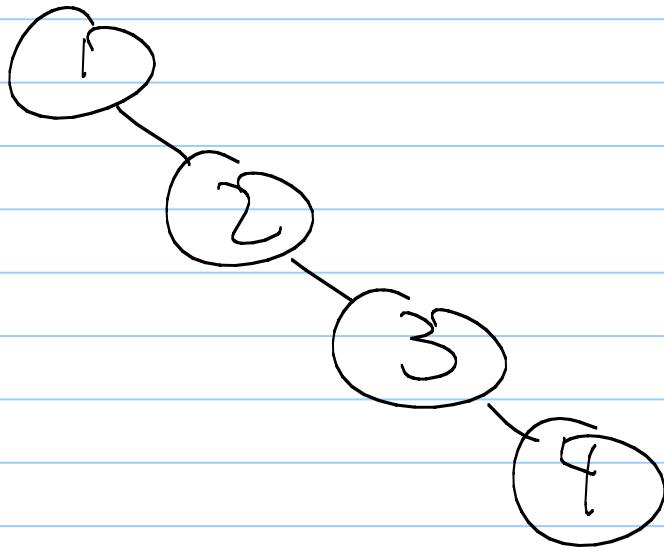


delete(16)

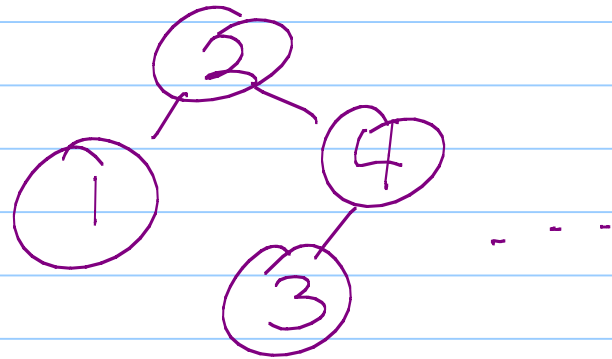
candidates to
replace:
neighbor
from inorder
traversal



Note: BSTs are not unique!



Can you make another BST with these elements?



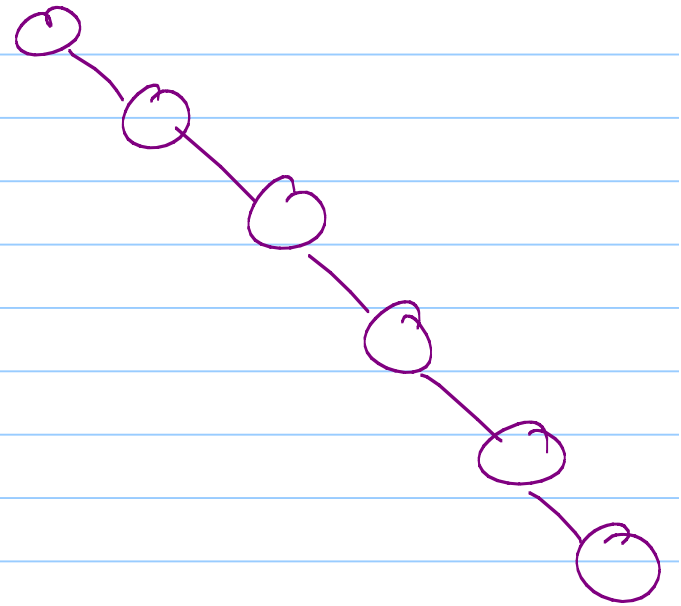
Runtime:

Find: $O(\text{depth tree})$

Insert: "

Delete: "

depth = $O(n)$



Code

- Will be pointer based. Why?
(could be unbalanced)

(Need nodes, iterators, etc.)

Today:

Code for generic binary trees.

BinaryTree.h will be generic -
not BSTs.

BST.h will inherit from BinaryTree.h
(but so will other classes.)

↳ AVL.h will balance binary search
trees.

