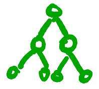
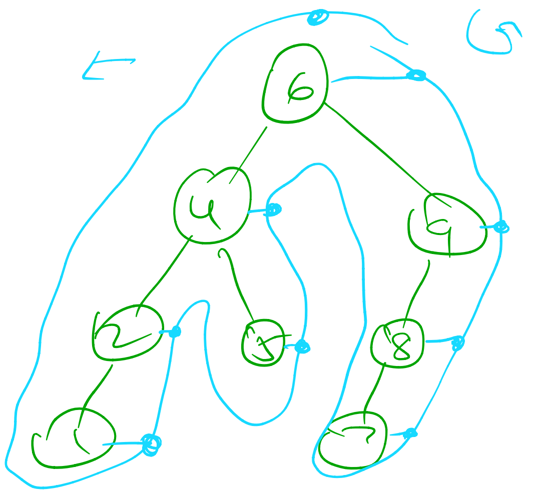
CS 61B Week 9: Trees



1. Tree-versal



a) Preorder:

b)

c)

d) $\underset{0}{\underline{6}}\ \underset{1}{\underline{4\ 9}}\ \underset{2}{\underline{2\ 5\ 8}}\ \underset{3}{\underline{1\ 7}}$

Depth First Search

* Preorder:
- doSomething() ←
- left() ←
- right()

(left)

* Postorder:
- left()
- right()
- doSomething() ←

(right)

* Inorder:
- left() ←
- doSomething()
- right()

(bottom)

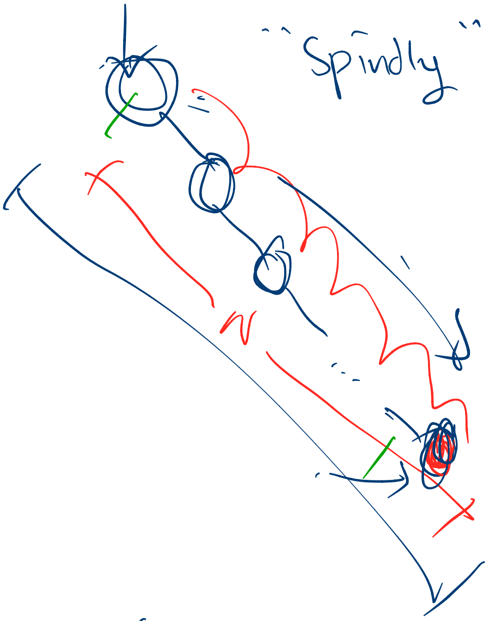
* BFS: Breadth First Search

"bushy"

2. Runtime

a) Best: $\Theta(1)$ DFS preorder

Worst: $\Theta(N)$



b) Best: $\Theta(1)$

Worst: $\Theta(N)$

getEl(0)

c) Best: $\Theta(1)$

Worst: $\Theta(N)$

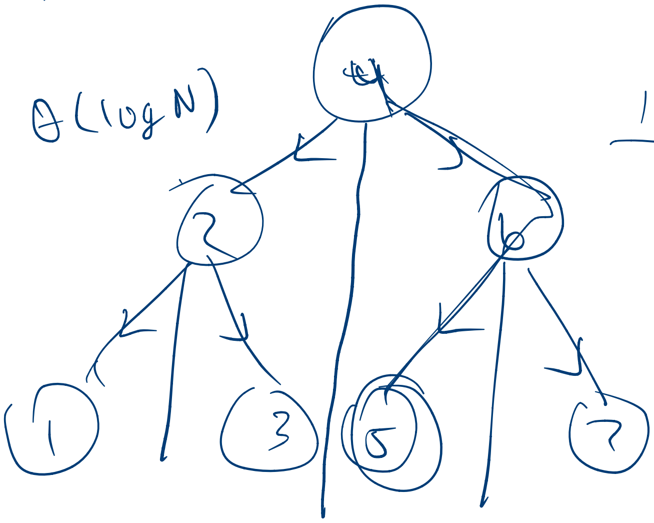
size()

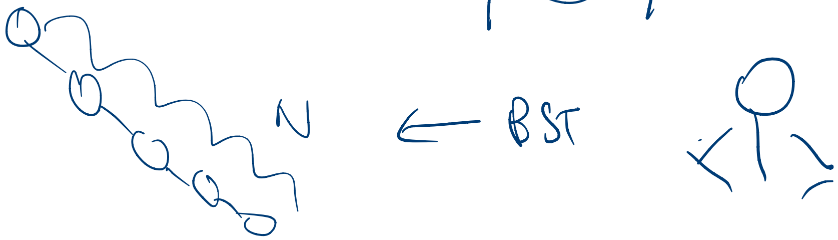
$\Theta(\log N)$

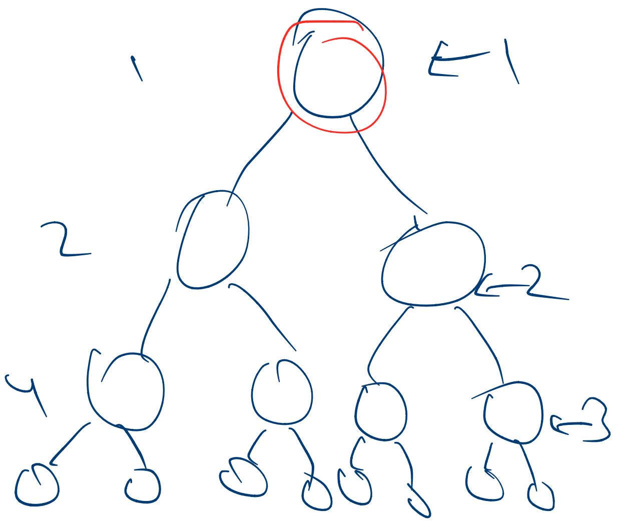
1...7

bushy

$\log N$







d levels

$$\underline{1+2+4}+8 = \underline{2^d - 1}$$

$$2^4 - 1 = 16 - 1 = 15$$

$$N = 2^d - 1$$

$$\log N = \log(2^d \cancel{-1})$$

$$\underline{2^d}\ \underline{-1}$$

$$2^d$$

$$\log N = \log 2^{d}$$

$$\log \underline{N} = \underline{d}\,\cancel{\log}\ \cancel{\log 2}$$

$$d = \log N$$

* for a bushy tree

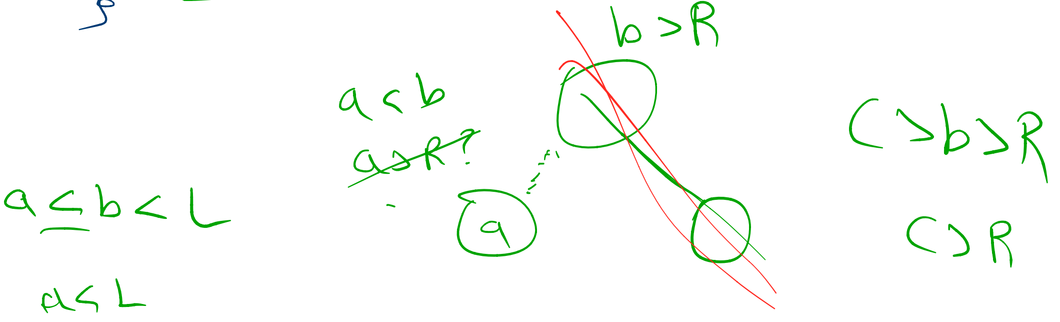
3. Pruning Trees

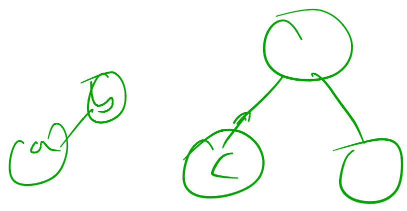
```
public BST pruneBST ( BST root, int L, int R) {
        if (root==null) {
                ret null;
        } else if ( root.label < L) {
                ret pruneBST ( root.right, L,R);
        } else if ( root.label > R) {
                ret pruneBST ( root.left, L,R);
        }
        root.left = pruneBST (root.left, L,R);
        root.right = pruneBST ( root.right, L,R);
        return root;
}
```

DFS

Preorder

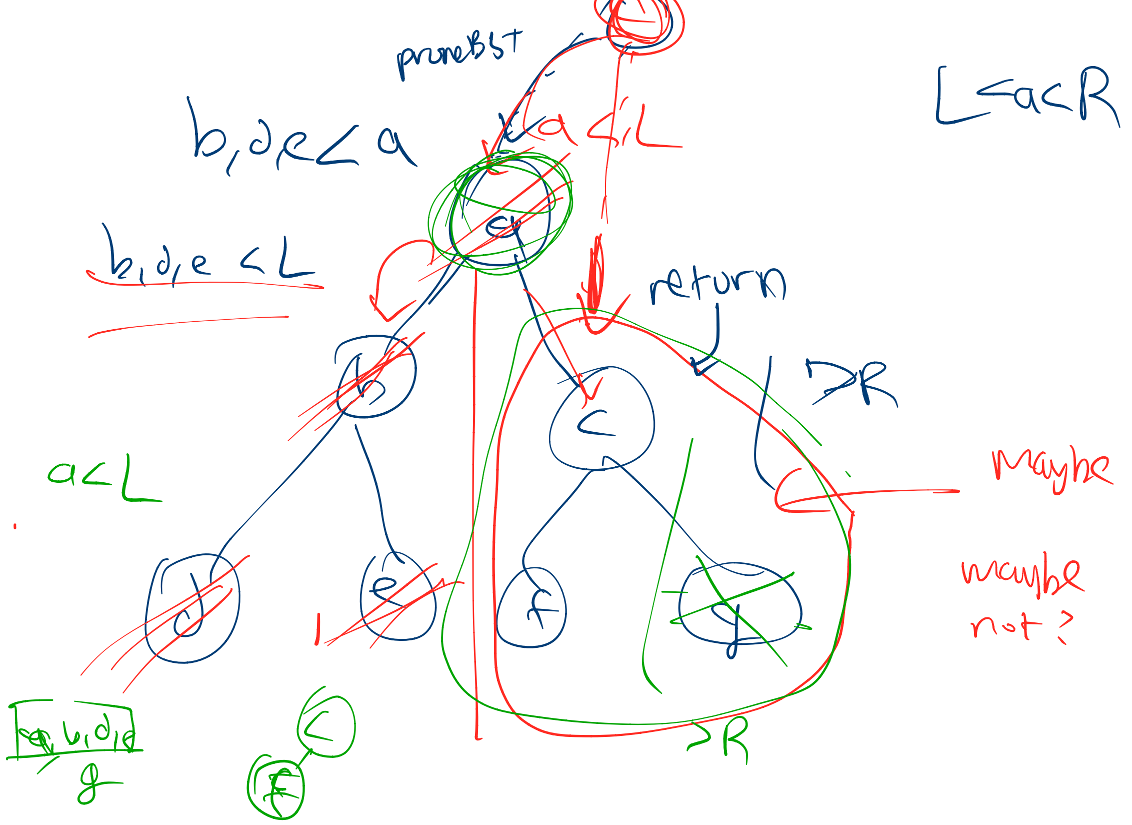
//pruning

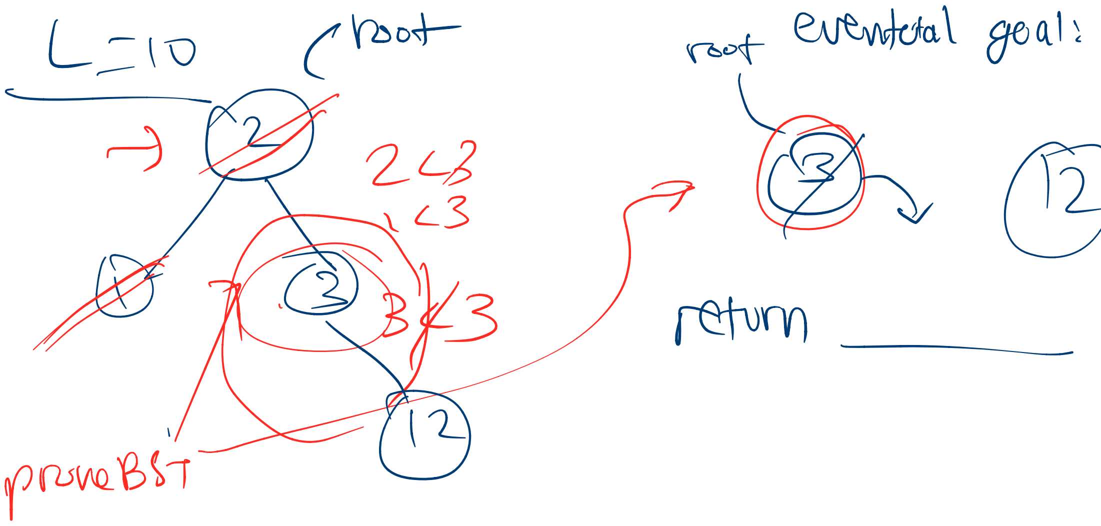


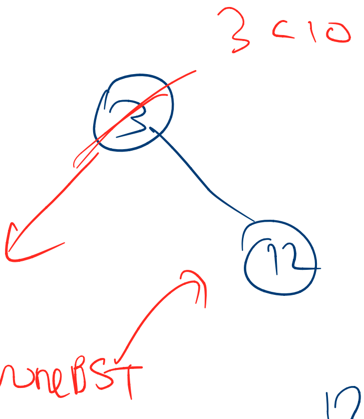


do something to the root

for b in branches(t):
    do something to child (b)


pruneBST
b,d,e < a
a ≤ L
L ≤ a < R
b,d,e < L
return
> R
maybe
maybe not?
a < L
a, b, d, e
g
> R


L = 10
root
2 < 3
1 < 3
3 < 3
pruneBST
root
eventual goal:
return



R = ∞