CS 225

Data Structures

September 19 – BST Remove

G Carl Evans

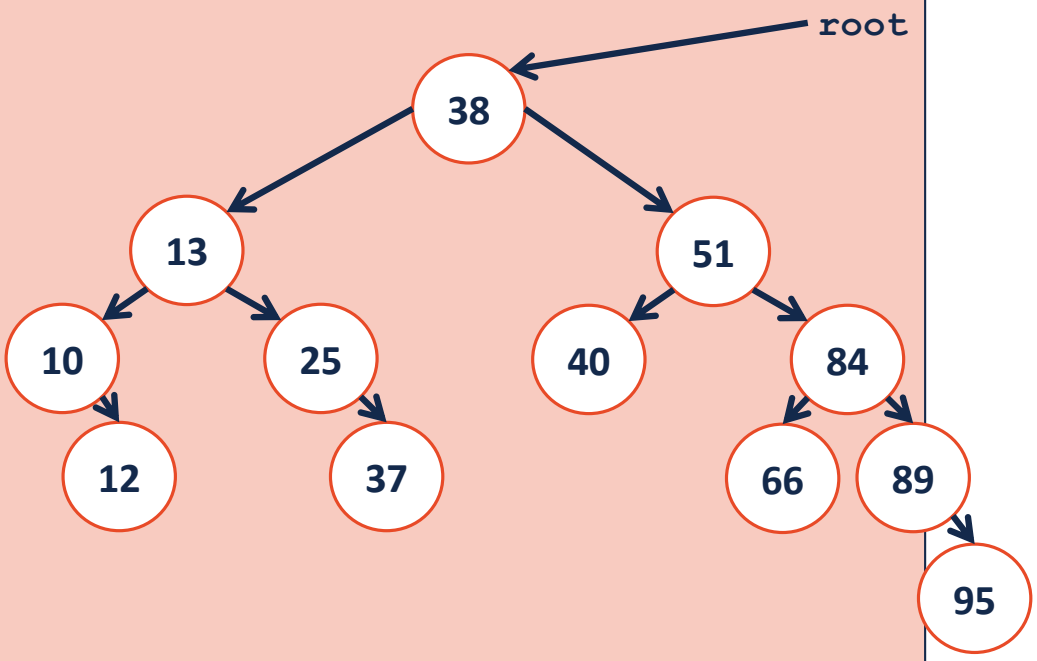
Exam 2 - Topics

- C++ review pointers and references
- Lists
  - Array with runtimes
  - Linked with runtimes
  - Stacks and Queues as special cases
- Trees
  - Terminology
  - Binary
  - BST

Website with more details and practice on PL by end of the day

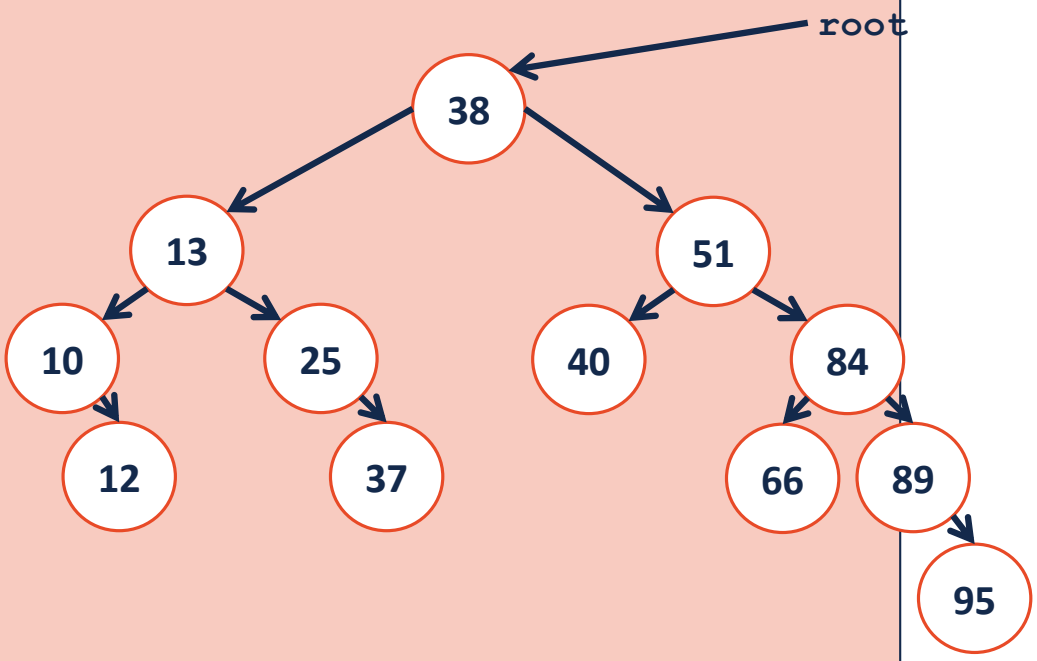
```
template<class K, class V>

TreeNode * &_find(TreeNode *& root, const K & key) {
```

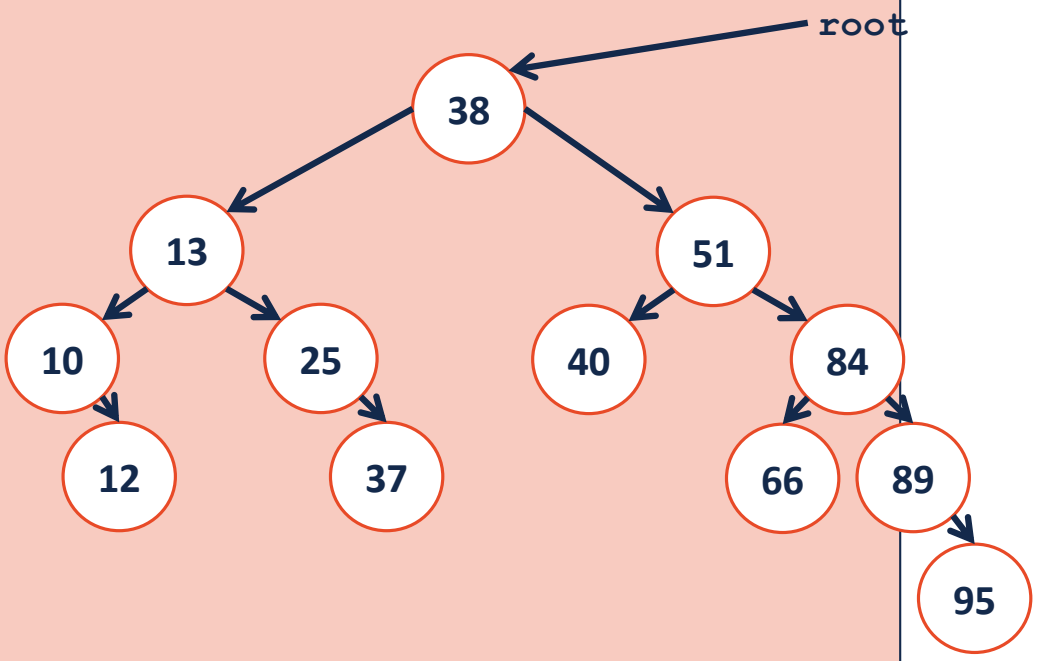


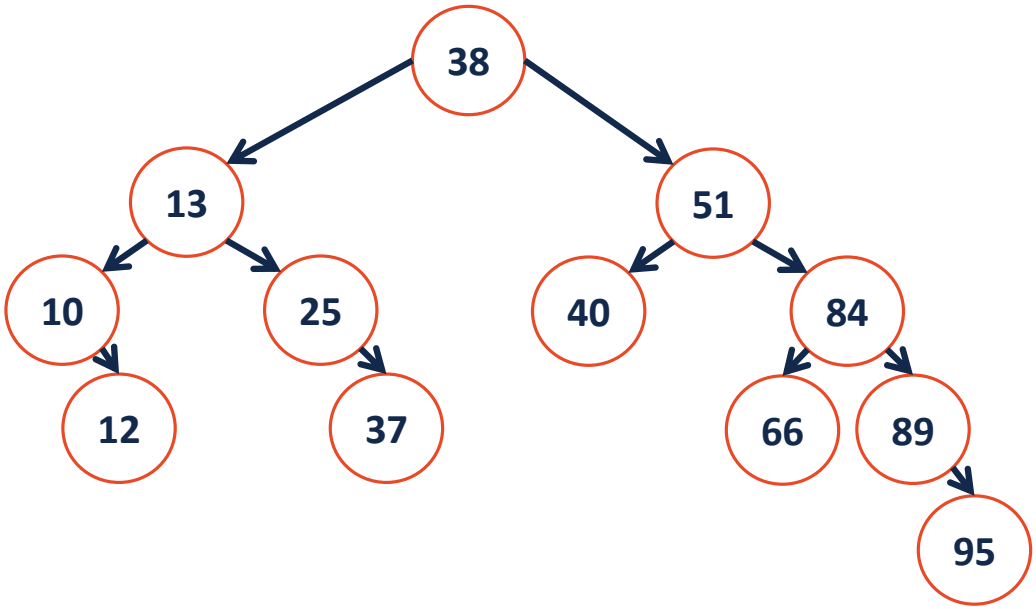
```
template<class K, class V>

void insert(TreeNode *& root, const K & key) {
```

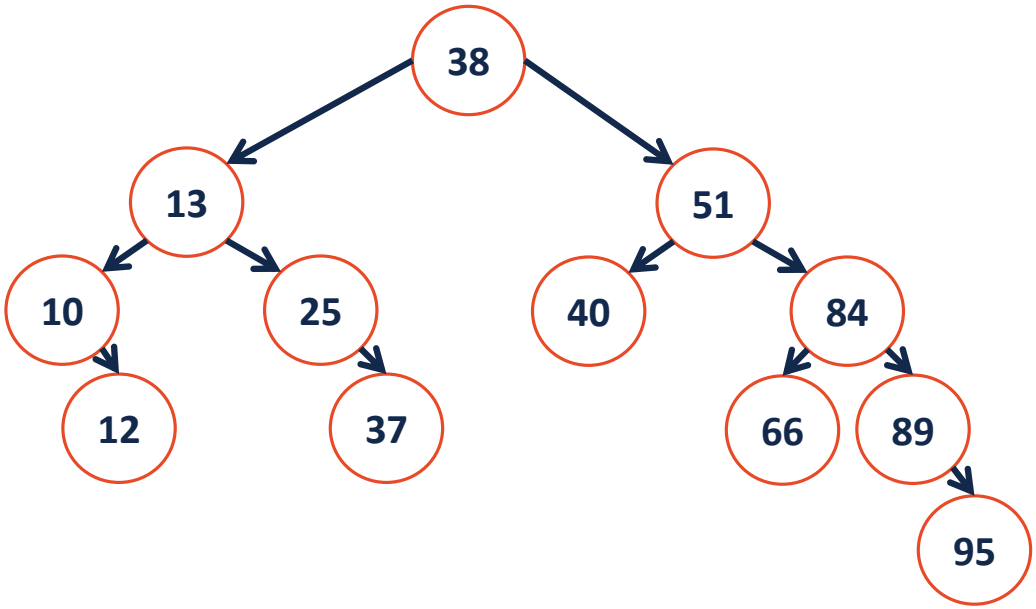


```
1  template<typename K, typename V>
2  _____________________________ _remove(TreeNode *& root, const K & key) {
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 }
```

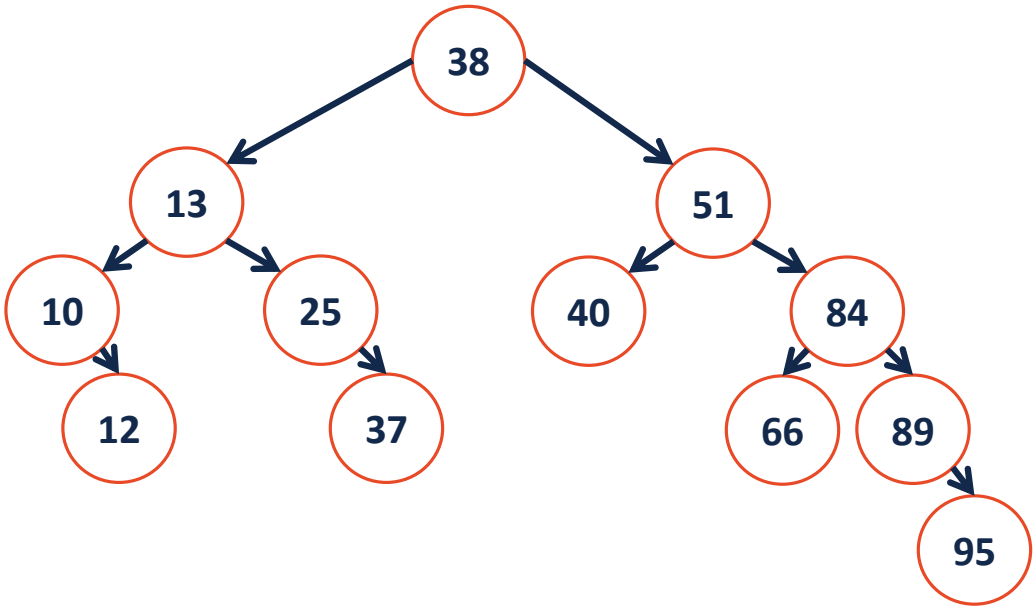




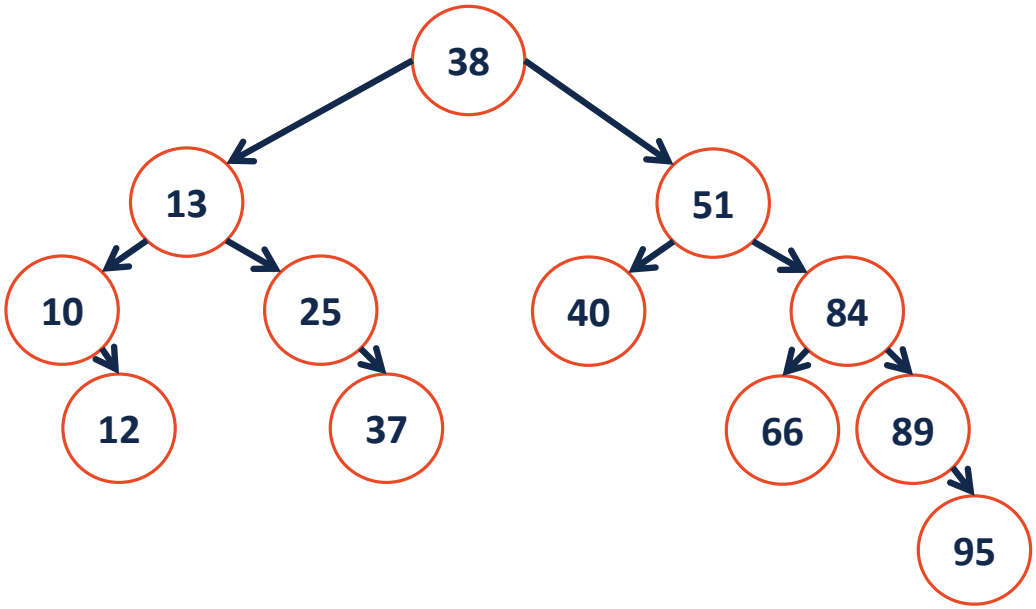

```
remove(40);
```




```
remove(25);
```




```
remove(10);
```




```
remove(13);
```

BST Analysis – Running Time

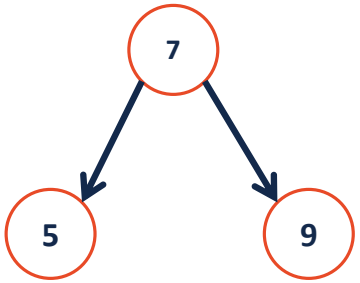
Operation	BST Worst Case
find	
insert	
delete	
traverse	

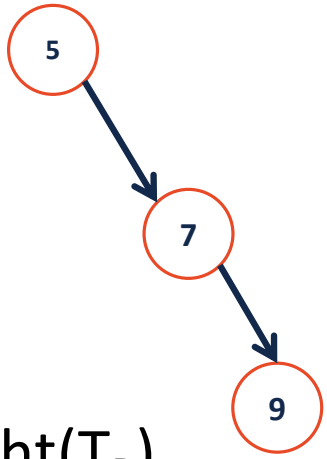
BST Analysis – Running Time

Operation	BST Average case	BST Worst case	Sorted array	Sorted List
find				
insert				
delete				
traverse				

Height-Balanced Tree

What tree makes you happier?





Height balance: $b = height(T_L) - height(T_R)$

A tree is height balanced if: