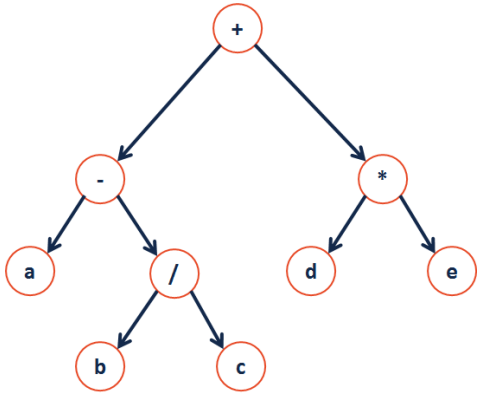
CS 225

#16: *Traversals and Dictionaries*

September 29, 2021 · *G Carl Evans*

A Different Type of Traversal

Strategy:



BinaryTree.cpp

```
void BinaryTree<T>::levelOrder(TreeNode * croot) {




}
```

Traversal vs. Search:

- **Traversal** visits every node in the tree exactly once.
- **Search** finds one (or more) element(s) in the tree.

Breadth First Traversal + Search:

Depth First Traversal + Search:

Runtime Analysis on a Binary Tree:

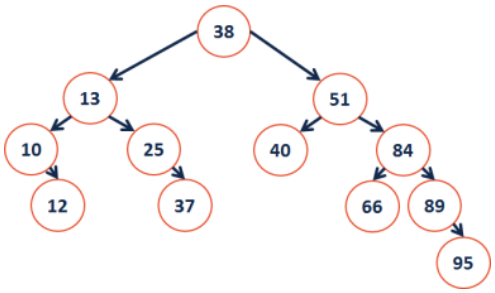
- Find an element: Best case? Worst case?
- Insertion of a sorted list of elements?
  Best case? Worst case?
- Running time bound by

Dictionary ADT

Dictionary.h

```
3
4  class Dictionary {
5    public:
6
7
8
9
10
11
12
13   private:
14
15
16 };
```

A Searchable Binary Tree?



Binary Search Tree Property:

Finding an element in a BST:

BST.hpp

```
template <typename K, typename V>
_____ find(const K & key) {



}

template <typename K, typename V>
_____ _find
    (TreeNode *& root, const K & key) {









}
```

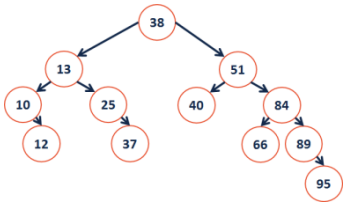
BST.cpp

```
template <class K, class V>
void BST::_insert(TreeNode *& root, K & key, V & value) {
  TreeNode * t = _find(root, key);
  t = new TreeNode(key, value);
}
```

Running time? _____ **Bound by?** _____

What happens when we run the bugged code above?

How do we fix the code?



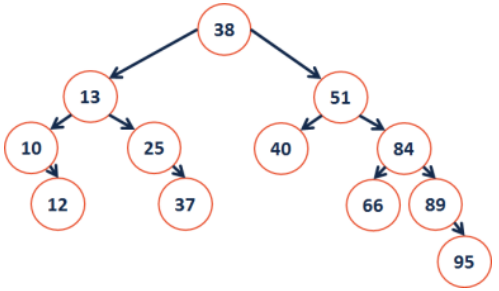
Removing an element from a BST:

`_remove(40)`

`_remove(25)`

`_remove(10)`

`_remove(13)`



CS 225 – Things To Be Doing:

1. Exam 1 on going CBTF/Online
2. lab_trees tomorrow
3. Daily POTDs