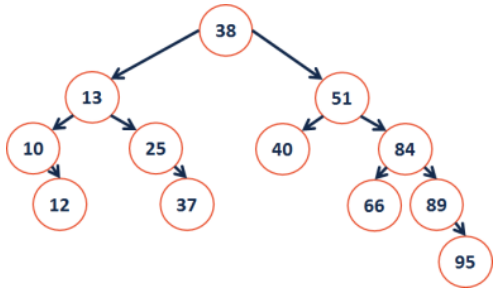
CS 225

# #18: BST Implementation

October 9, 2017

## Binary Search Tree (BST)



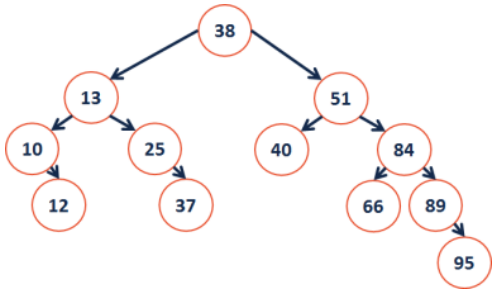

## Tree Analysis:

- **Best case?**
- **Worst case?**
- **Insertion of a sorted list of elements?**
- **Running time?**

**BST.h**

```
1  #ifndef DICTIONARY_H
2  #define DICTIONARY_H
3
4  template <class K, class V>
5  class BST {
6    public:
7      BST();
8      void insert(const K key, V value);
9      V remove(const K & key);
10     V find(const K & key) const;
11     TreeIterator traverse() const;
12   private:
13
14
15
16
17
18 };
19 #endif
```

## Finding an element in a BST:



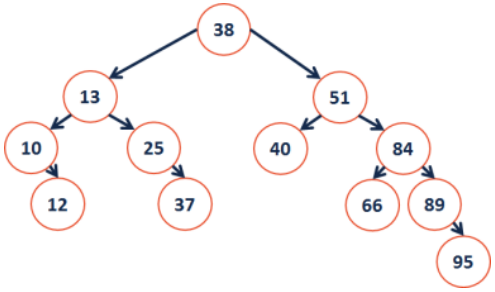

**BST.cpp**

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




}
```

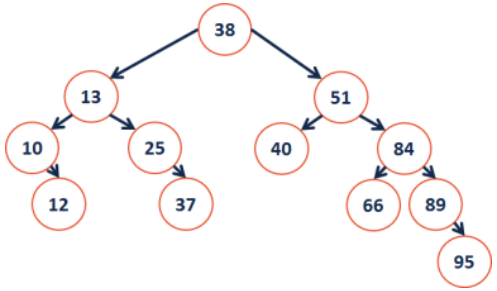
**...running time: _____**

---

**What if we did <u>not</u> pass a pointer by reference?**



**Inserting an element into a BST:**




**BST.cpp**

```
template <class K, class V>
void BST::_insert(TreeNode *& root, K key, V value) {




}
```

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

---

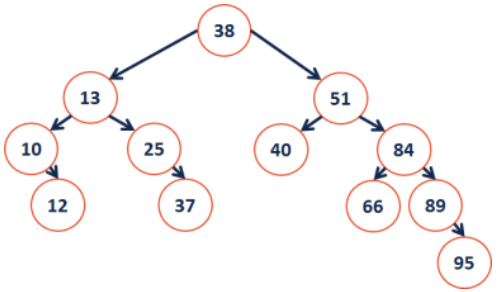
**Removing an element from a BST:**

`_remove(40)`

`_remove(25)`

`_remove(10)`

`_remove(13)`




| One-child Remove | Two-child remove |
|---|---|
| | |

**BinaryTree.cpp**

```
template <class K, class V>
void BST::_remove(TreeNode *& root, const K & key) {




}
```

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

**CS 225 – Things To Be Doing:**

1. Exam #5 live now! (Theory Exam: lists, stacks, queues)
2. MP3 due today, Monday, October 9, 2017; MP4 out tomorrow
3. lab_trie coming up in lab sections
4. Daily POTDs