# 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

```cpp
#ifndef DICTIONARY_H
#define DICTIONARY_H

template <class K, class V>
class BST {
public:
    BST();
    void insert(const K key, V value);
    V remove(const K & key);
    V find(const K & key) const;
    TreeIterator traverse() const;
private:
};

#endif
```

Finding an element in a BST:

```cpp
BST.cpp

// find
(TreeNode *& root, const K & key) const {

...running time: __________

What if we did not 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:

```
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