#16: Binary Search Tree (BST)  
March 8, 2021 · G Carl Evans

## 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

```cpp
class Dictionary {
public:
private:
};
```

## A Searchable Binary Tree?

### BST.hpp

```cpp
template <typename K, typename V>
token find(const K & key) const {
}
token find(TreeNode *& root, const K & key) const {
}
```

## Finding an element in a BST:

## Inserting an element into a BST:
Removing an element from a BST:

```cpp
def _remove(40):
    _remove(25):
    _remove(10):
    _remove(13)
```

<table>
<thead>
<tr>
<th>One-child Remove</th>
<th>Two-child remove</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Running time? ____________    Bound by? ___________

What if we did not pass a pointer by reference?

Running time? ____________    Bound by? ___________

**CS 225 – Things To Be Doing:**

1. mp_lists due today
2. Upcoming Lab: lab_huffman
3. Daily POTDs