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

Finding an element in a BST:

```cpp
# Dictionary ADT

Dictionary.h

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

BST.hpp

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

}

template <typename K, typename V>
______________ find
    (TreeNode * & root, const K & key)
    const {
};
```
Inserting an element into a BST:

```
// BST.hpp
template <typename K, typename V>
void BST<K, V>::_insert(TreeNode *& root, K key, V value) {
}
```

Running time? ____________ Bound by? ____________

What if we did not pass a pointer by reference?

Removing an element from a BST:

```
_remove(40)
_remove(25)
_remove(10)
_remove(13)
```

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

Running time? ____________ Bound by? ____________

---

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

---

```
// BST.hpp
template <typename K, typename V>
void BST<K, V>::_insert(TreeNode *& root, K key, V value) {
}
```

Running time? ____________ Bound by? ____________

---

CS 225 – Things To Be Doing:

1. Theory Exam 2 Topics List Posted (exam next week)
2. MP3 extra credit on-going; MP3 due Monday, Oct. 8
3. Upcoming Lab: lab_trees
4. Daily POTDs