CS 225
Data Structures

February 7 – Trees Theory and Traversal
G Carl Evans
How many NULLs?

**Theorem:** If there are $n$ data items in our representation of a binary tree, then there are ___________ NULL pointers.
How many NULLs?

Base Cases:

n = 0:

n = 1:

n = 2:
How many NULLs?

Induction Hypothesis:
How many NULLs?

Consider an arbitrary tree $T$ containing $n$ data elements:
Access All the Nodes - Traversals
template<class T>
void BinaryTree<T>::__Order(TreeNode * cur) {
}
```cpp
template<class T>
void BinaryTree<T>::___Order(TreeNode * cur) {
    if (cur != NULL) {
        ______________;
        ___Order(cur->left);
        ______________;
        ___Order(cur->right);
        ______________;
    }
}
```
Traversals

```cpp
template<class T>
void BinaryTree<T>::___Order(TreeNode * cur) {
    if (cur != NULL) {
        ____________________;
        ___Order(cur->left);
        ____________________;
        ___Order(cur->right);
        ____________________;
    }
}
```
A Different Type of Traversal
A Different Type of Traversal

template<class T>
void BinaryTree<T>::levelOrder(TreeNode * root) {

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

1 template<class T>
2 void BinaryTree<T>::levelOrder(TreeNode * root) {

1 +
2 -
3 /
4 *
5 a
6 b
d
c
e

10
11
12
13
14
15
16
17
Traversals vs. Search

Traversals

Search
Search: Breadth First vs. Depth First

Strategy: Breadth First Search (BFS)

Strategy: Depth First Search (DFS)
Dictionary ADT

Data is often organized into key/value pairs:

- UIN ➔ Advising Record
- Course Number ➔ Lecture/Lab Schedule
- Node ➔ Incident Edges
- Flight Number ➔ Arrival Information
- URL ➔ HTML Page

...
#pragma once

class Dictionary {
  public:
  // ...

  private:
  // ...
};
Binary Tree as a Search Structure
Binary Tree as a Search Structure
Binary _______________ Tree (BST)

A **BST** is a binary tree $T$ such that: