CS 225
Data Structures

Feb. 22 – BST Remove
Wade Fagen-Ulmschneider, Craig Zilles
template<typename K, typename V>
void BST::_insert(TreeNode * & root, K & key, V & value) {
    TreeNode * t = _find(root, key);
    t = new TreeNode(key, value);
}
template<typename K, typename V>
void BST::_insert(TreeNode * & root, K & key, V & value) {
    TreeNode * t = _find(root, key);
    t = new TreeNode(key, value);
}
template<typename K, typename V>

void remove(TreeNode *& root, const K & key) {

}
remove(40);
remove(25);
remove(10);
remove(13);
## BST Analysis – Running Time

<table>
<thead>
<tr>
<th>Operation</th>
<th>BST Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>find</td>
<td></td>
</tr>
<tr>
<td>insert</td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
</tr>
<tr>
<td>traverse</td>
<td></td>
</tr>
</tbody>
</table>
BST Analysis

Every operation that we have studied on a BST depends on the height of the tree: $O(h)$.

...what is this in terms of $n$, the amount of data?

We need a relationship between $h$ and $n$:

$f(h) \leq n \leq g(h)$
BST Analysis

**Q: What is the maximum number of nodes in a tree of height \( h \)?**
BST Analysis

**Q:** What is the minimum number of nodes in a tree of height $h$?

What is the maximum height for a tree of $n$ nodes?
BST Analysis

Therefore, for all BST:

**Lower bound:**

**Upper bound:**
BST Analysis

The height of a BST depends on the order in which the data is inserted into it.

ex: 1 3 2 4 5 7 6  vs.  4 2 3 6 7 1 5

Q: How many different ways are there to insert keys into a BST?

Q: What is the average height of all the arrangements?
BST Analysis

Q: How many different ways are there to insert keys into a BST?

Q: What is the average height of all the arrangements?
## BST Analysis – Running Time

<table>
<thead>
<tr>
<th>Operation</th>
<th>BST Average case</th>
<th>BST Worst case</th>
<th>Sorted array</th>
<th>Sorted List</th>
</tr>
</thead>
<tbody>
<tr>
<td>find</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>insert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>traverse</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Height-Balanced Tree

What tree makes you happier?

Height balance: \[ b = \text{height}(T_L) - \text{height}(T_R) \]

A tree is height balanced if: