What happens when we run the bugged code above?

Insert

<table>
<thead>
<tr>
<th>BST.cpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>template &lt;class K, class V&gt; void BST::insert(TreeNode * &amp; root, K &amp; key, V &amp; value) { TreeNode * t = _find(root, key); t = new TreeNode(key, value); }</td>
</tr>
</tbody>
</table>

How do we fix the code?

Removing an element from a BST:

- remove(40)
- remove(25)
- remove(10)
- remove(13)

One-child Remove

<table>
<thead>
<tr>
<th>BinaryTree.cpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>template &lt;class K, class V&gt; void BST::_remove(TreeNode * &amp; root, const K &amp; key) { }</td>
</tr>
</tbody>
</table>

Two-child remove

BST Analysis:

Every operation we have studied on a BST depends on:

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

BST – Simple Proofs

Q: Given a height h, what is the maximum number of nodes (n) in a valid BST of height h? Provide an outline of a proof.
Q: Given a height \( h \), what is the minimum number of nodes \( n \) in a valid BST of height \( h \)? Provide an outline of a proof.

**Final BST Analysis**
For every height-based algorithm on a BST:

**Lower Bound:**

**Upper Bound:**

Why use a BST over a linked list?

---

Q: How does our data determine the height?

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

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

Q: What is the average height of every arrangement?

...what is the intuition here?

**Height Balance on BST**

What tree makes you happier?

We define the **height balance** \( b \) of a BST to be:

We define a BST tree \( T \) to be **height balanced** if:

---

**CS 225 – Things To Be Doing:**

1. mp_mosaic released ec due Monday.
2. lab_quacks starts today in lab
3. exam 1 reschedule window Saturday 2/26 – Monday 2/28.
4. Daily POTDs