template<class K, class V>

TreeNode* _remove(TreeNode* & root, const K & key) {
}

1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
template<class K, class V>

TreeNode *\& root,
const K & key) {
remove(40);
remove(25);
remove(10);
remove(13);
BST Analysis

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

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

We need a relationship between \( h \) and \( n \):

\[
    h \geq f(n)
\]

\[
    h \leq g(n)
\]
Q: What is the maximum number of nodes in a tree of height $h$?
Q: What is the minimum number of nodes in a tree of height $h$?

What is the greatest possible 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 – 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:
Exam 5 (Theory) is ongoing!
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP4: Available later today!
*Due: Monday, Oct. 23 at 11:59pm*

Lab!
*Due: Sunday, Oct. 15 at 11:59pm*

POTD
Every Monday-Friday – *Worth +1 Extra Credit /problem (up to +40 total)*