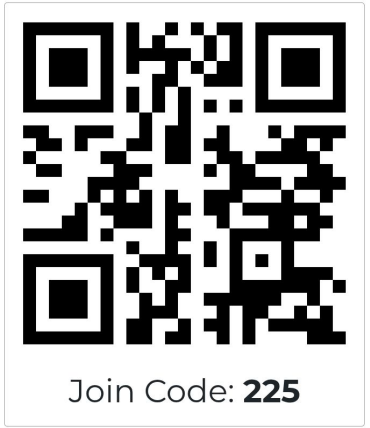
Announcements

1. MP Mosaics EC due Today!
2. Exam 2 Grades In Process
3. Extra Credit Survey due Wednesday
    a. 10 more people to hit threshold!

Extra Credit Survey!



Warm-Up Question: What is the minimum number of nodes needed to make an AVL tree with height 3?

AVL Tree Analysis

Learning Objectives

1. Prove the upper bound on an AVL Tree's Height
2. Enumerate the runtime of Insert, Find, Remove, and Traverse for every Data Structure Seen so far

AVL Tree Analysis

We know: insert, remove and find runs in: _____.

We will show that: h is _____.

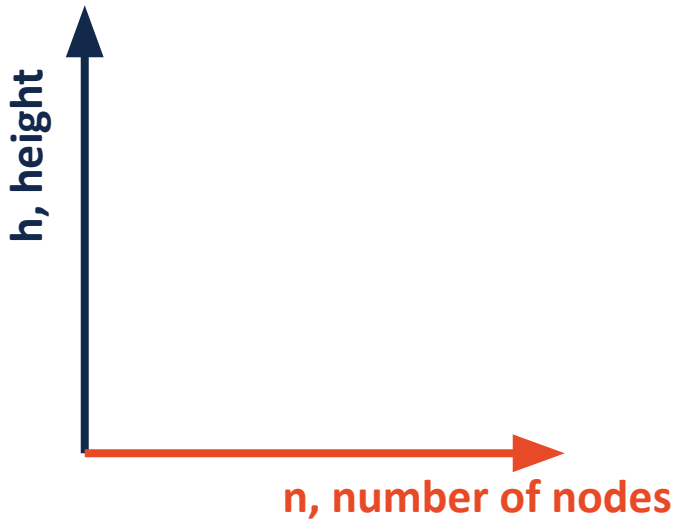
Goal: Find the lowest upper bound, big O, on the maximum height of an AVL tree given the number of nodes.

Big O - Lowest Upper Bound

Definition of big-O: Given two functions **f(n)** and **g(n)**, we say that **f(n)** is **O(g(n))** if there exist constants **c > 0** and $\mathbf{n_0}$ **>=** 0

such that **f(n) <= c*g(n)** for all **n >=** $\mathbf{n_0}$.

…or, with pictures:

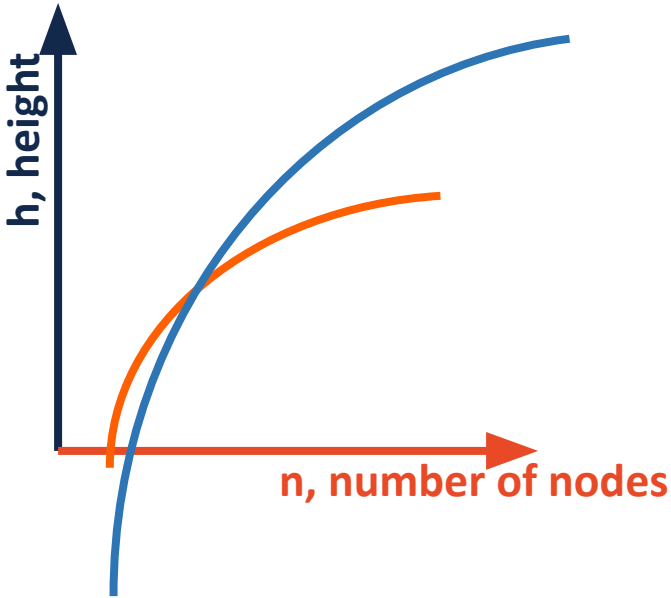


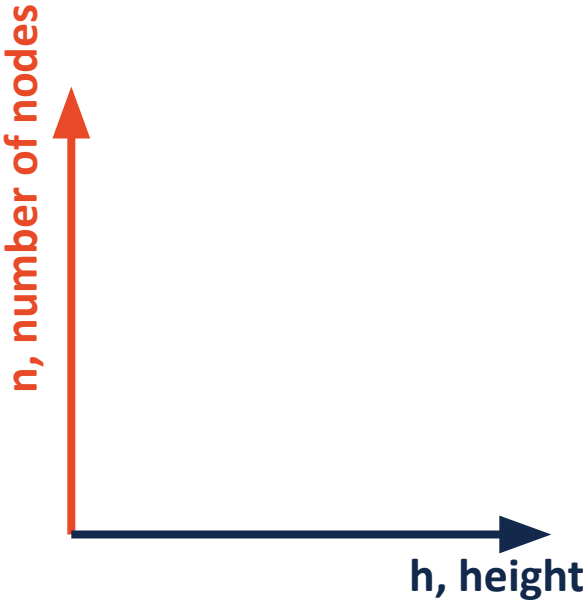
New Goal

Goal: Find the lowest upper bound, big O, on the maximum height of an AVL tree given the number of nodes.

New Goal: Find the highest lower bound on the minimum number of nodes in an AVL tree given the height, N(h)

Invert and Find Highest Lower Bound





Small Examples

h = 0 h = 1 h = 2 h = 3 h = 4

Recurrence

$$N(h) = 1 + N(h - 1) + N(h - 2)$$

Conclusion

$h = O(\log(n))$

Summary

AVL Trees

Max height: 1.44 * log(n)

Rotations:

Red-Black Trees

Max height: 2 * log(n)

Constant number of rotations on insert, remove, and find

	Unsorted Array	Sorted Array	Unsorted List	Sorted List	Binary Tree	BST	AVL
Find							
Insert							
Remove							
Traverse							

	Unsorted Array	Sorted Array	Unsorted List	Sorted List	Binary Tree	BST	AVL
Find	O(n)	O(log n)	O(n)	O(n)	O(h)/O(n)	O(h)/O(n)	O(h)/O(log n)
Insert	O(n)/O*(1)	O(n)	O(1)	O(n)	O(h)/O(n)	O(h)/O(n)	O(h)/O(log n)
Remove	O(n)	O(n)	O(n)	O(n)	O(h)/O(n)	O(h)/O(n)	O(h)/O(log n)
Traverse	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)