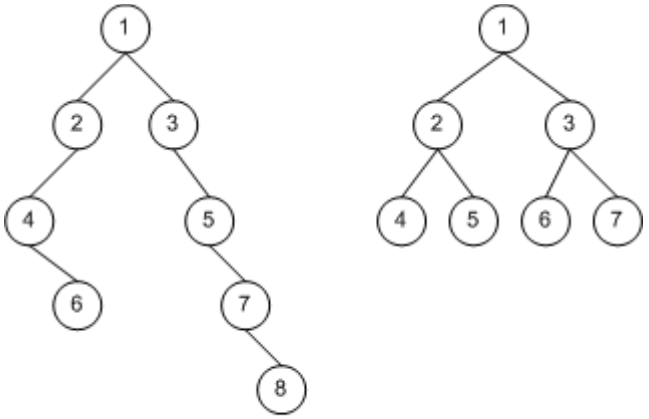

(a)

(b)

CSSE 230 Day 10

Size vs height in a Binary Tree

After today, you should be able to…
… use the relationship between the size and height of a tree to find the maximum and minimum number of nodes a binary tree can have
…understand the idea of mathematical induction as a proof technique

http://i.msdn.microsoft.com/dynimg/IC71494.gif

Announcements

- Today:
  - Size vs height of trees: patterns and proofs
  - Q/A and worktime for BSTs

- Due after that:
  - Displayable Binary Tree
  - Meet partner today

Questions?

Size and Height of Binary Trees

- Notation:
  - Let T be a tree
  - Write h(T) for the height of the tree, and
  - N(T) for the size (i.e., number of nodes) of the tree

- Given h(T), what are the bounds on N(T)?
  - N(T) <= _____ and N(T) >= _____

- Given N(T), what are the bounds on h(T)?
  - Solve each inequality for h(T) and combine

Extreme Trees

- A tree with the maximum number of nodes for its height is a **full** tree.
  - Its height is O(log N)
- A tree with the minimum number of nodes for its height is essentially a _____ _____
  - Its height is O(N)
- Height matters!
  - Recall that the algorithms for search, insertion, and deletion in a binary search tree are O(h(T))

To prove recursive properties (on trees), we use a technique called mathematical induction

- Actually, we use a variant called *strong induction* :

The former governor of California

Strong Induction

- To prove that p(n) is true for all n >= n_0:
  - Prove that p(n_0) is true (base case), and
  - For all k > n_0, prove that if we assume p(j) is true for $n_0 \leq j < k$, then p(k) is also true

- An analogy for those who took MA275:
  - Regular induction uses the previous domino to knock down the next
  - Strong induction uses all the previous dominos to knock down the next!

- Warmup: prove the arithmetic series formula
- Actual: prove the formula for N(T)

Current assignment

Questions and answers

Meet partner

Worktime

Implementation issues, part 2

- Modifying (inserting/deleting) from a tree should cause any current iterators to fail (throw a ConcurrentModificationException).
  - How do you detect this?
- How do you remove from an iterator?
  - Just call BST remove().
  - But throw exceptions if next() hasn't been called, or if remove is called twice in a row. (Javadoc for TreeSet iterator has details.)