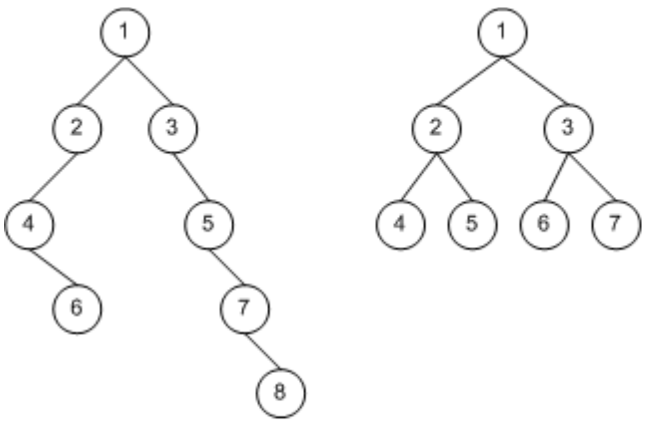
Q1



(a) (b)

CSSE 230 Day 11

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

Term project starts Day 13

Preferences for partners for the term project (groups of 3)

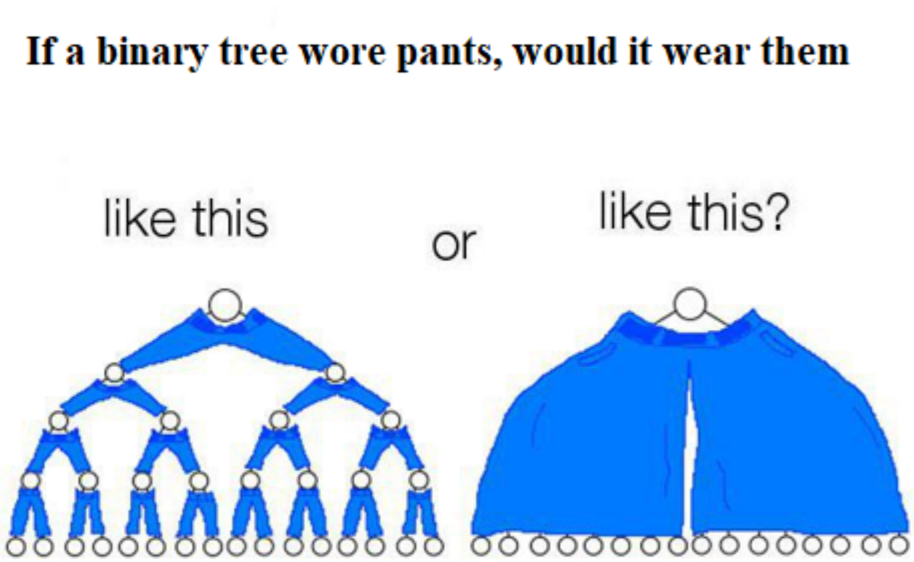
Partner preference survey on Moodle – Day 11

- Preferences balanced with experience level + work ethic
  - If course grades are close, I'll honor "prearranged teammate" preferences
  - If no "prearranged teammate", best to list several potential members
  - If you don't want to work with someone, that preference will be honored
  - Historical evidence indicates working with others in a similar current CSSE230 grade attainment level often pans out best

Some questions you might consider asking potential programming partners:

- What final grade range are you aiming for in CSSE230?
- Do you like to get it done early or to procrastinate?
- Do you prefer to work daytime, evening, late night?
- Do you normally get a lot of help on the homework?
- Survey is due Fri Jan 10, 5 PM – do it as soon as you can

Some meme humor



http://www.smosh.com/smosh-pit/memes/internets-best-reactions-if-dog-wore-pants
http://knowyourmeme.com/photos/1272773-if-a-dog-wore-pants

Other announcements

- Today:
  - Size vs height of trees: patterns and proofs
- Wrapping up the BST assignment, and worktime.

Extreme Trees

- A tree with the maximum number of nodes for its height is a **full** tree.
- full binary tree – each non-leaf node has exactly two children, all leaves at same level.

- A tree with the minimum number of nodes for its height is called **degenerate**

- Height matters!
  - Recall that the algorithms for search, insertion, and deletion in a binary search tree are **O(h(T))**

Proving a Universal Statement

- Example:

  Open statement $S(n)$

  For all integers $n \geq 0$, $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

- "for all" (∀) is called the universal qualifier

- How to prove?
  - Can't do it one-by-one: there are infinitely many statements to prove!
  - Could try direct proof: show $\forall n\, S(n)$
    - Typically, pick *arbitrary but specific n* and prove using logic
  - But, often easier to use induction: show $S(0)$ and

$$\forall k\ (S(k) \rightarrow S(k+1))$$

Q5

Mathematical Induction

To prove that P(n) is true for all $n \geq n_0$:

1. Basis step: Prove that $P(n_0)$ is true (base case), and
2. Induction step: Prove that **if P(k) is true** for any $k \geq n_0$, then P(k+1) is also true.

    [This part of the proof must work for all such k!]

$$\left(P(n_0)\ \&\ \forall k\left(P(k) \rightarrow P(k+1)\right)\right) \rightarrow \forall n\, P(n)$$

- Note: we still need to prove a universal statement! But the advantage is that we're allowed to assume the induction hypothesis (truth of the "previous case" $P(k)$) in proving the "next case" $P(k+1)$.

- Example: prove the arithmetic sum formula

Strong Induction

$$\Big(P(n_0)\ \&\ \forall k\big((P(n_0)\ \&\cdots\&\ P(k)) \to P(k+1)\big)\Big) \to\ \forall n\ P(n)$$

- Strengthen the induction hypothesis
- Rather than assume truth of just the previous case $P(k)$, assume truth of all previous cases $P(n_0), P(n_0+1), P(n_0+2), \dots, P(k)$.

Q6

Strong Induction

- To prove that p(n) is true for all $n \geq n_0$:
  - 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:
  - Regular induction uses the previous domino to knock down the next
  - Strong induction uses all the previous dominos to knock down the next!
- Example: prove the upper bound on N(T) in terms of h(T)