CS 561: Homework 1

Prof. Jared Saia, University of New Mexico

Remember: You are encouraged to work on the homework in groups, but please observe the "Star Trek" rule from the syllabus.

1. Prove that
$$\log(n!) = \Theta(n \log n), \qquad n! = \omega(2^n), \qquad \text{and} \qquad n! = o(n^n).$$

2. Let f and g map the integers to the positive real numbers, and assume that $f(n) = O(g(n))$. For each statement below, determine whether it is true or false, and give either a proof or a counterexample.

   (a) $\log_2 f(n) = O(\log_2 g(n))$.

   (b) $2^{f(n)} = O(2^{g(n)})$.

   (c) $f(n)^2 = O(g(n)^2)$.

   (d) If $f(n) = o(g(n))$, then $\log_2 f(n) = o(\log_2 g(n))$.

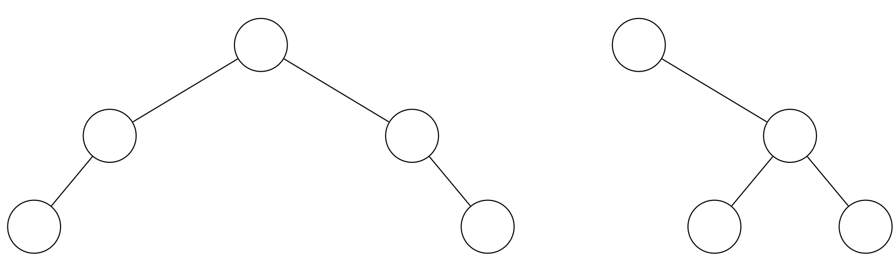
3. Let a_n be the number of binary strings of length n that do not contain two consecutive 1's. The empty string is the unique such string of length 0.

   (a) Derive and justify a recurrence for a_n, including base cases.

   (b) Let $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. Prove by induction that, for all $n \geq 0$,
   $$a_n = F_{n+2}.$$

   (c) Prove, using your recurrence, that
   $$2^{\lfloor n/2 \rfloor} \leq a_n \leq 2^n.$$

4. In a *leaf-balanced binary tree*, every node with two children has the same number of leaves in the subtrees rooted at its two children. A perfect binary tree is leaf-balanced; the following are two other examples.



   Let n be the number of nodes in the tree. Prove by induction on n that the number of leaves in a leaf-balanced binary tree is always a power of 2.

   Hint: Apply the inductive hypothesis to the subtree or subtrees rooted at the children of the root.