CS 362, HW 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 $\log n! = \Theta(n \log n)$ and that $n! = \omega(2^n)$ and $n! = o(n^n)$

2. Assume you have functions f and g, such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether you think it is true or false and give either a proof or a counterexample

   (a) $\log_2 f(n)$ is $O(\log_2 g(n))$

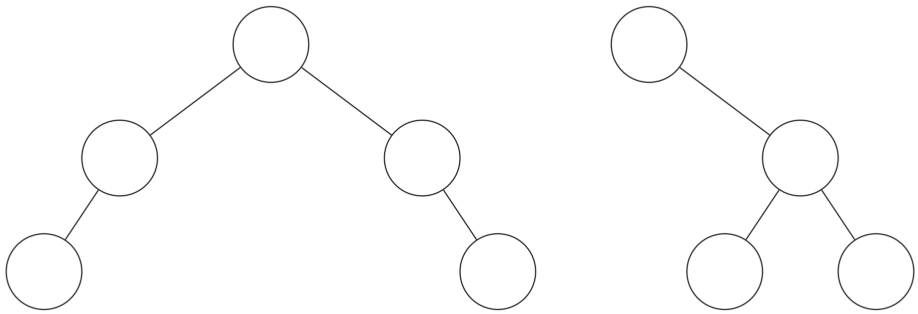
   (b) $2^{f(n)}$ is $O(2^{g(n)})$

   (c) $f(n)^2$ is $O(g(n)^2)$

3. Write and solve a recurrence relation giving the number of strings of n digits containing at least one 9. For example, if $n = 5$, then 02309 would be one such string.
   In particular, let $f(n)$ be the number of strings of n digits with at least one 9. First, write an equation $f(n) = ***$, where the *** part contains smaller sub-problems, i.e. the $f(j)$ terms all have $j < n$. Then give a base case for the recurrence. Finally, use guess and check to solve the recurrence to within $\Theta()$ bounds.

4. In a *leaf-balanced binary tree*, any node with 2 children has the same number of leaves in the sub-trees rooted at both children.

   A full binary tree is leaf-balanced; below are two other examples.



Let n be the number of nodes in the tree. Prove by induction on n that the number of leaf nodes in a leaf-balanced binary tree is always a power of 2. Hint: Apply the IH to subtree(s) of the root node.