Section Handout 2

Problem One: Prime Numbers

A natural number $p \geq 2$ is called ***prime*** iff it has no positive divisors except 1 and itself. A natural number is called ***composite*** iff it is the product of two natural numbers m and n, where both m and n are greater than one. Prove, by strong induction, that every natural number greater than one can be written as a product of prime numbers.

Problem Two: Picking Coins

Consider the following game for two players. Begin with a pile of n coins for some $n \geq 0$. The first player then takes between one and ten coins out of the pile, then the second player takes between one and ten coins out of the pile. This process repeats until some player has no coins to take; at this point, that player loses the game. Prove that if the pile begins with a multiple of eleven coins in it, the second player can always win.

Problem Three: Factorials! Multiplied together!

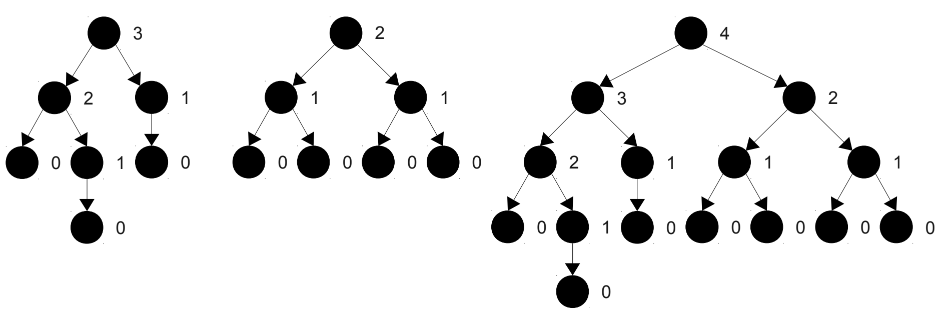
Prove by induction that for any $m, n \in \mathbb{N}$, that $m!n! \leq (m + n)!$

Problem Four: AVL Trees

An AVL tree is a tree structure defined recursively:

- An AVL tree of order 0 is a single node.
- An AVL tree of order 1 is either a single node with an AVL tree of order 0 as a child or a single node with two AVL trees of order 0 as children.
- For any $n \in \mathbb{N}$, an AVL tree of order $n + 2$ is either a single node with two children, which are AVL trees of order $n + 1$, or a single node with two children, one of which is an AVL tree of order n and one of which is an AVL tree of order $n + 1$.

For example, the following are AVL trees. Each node is annotated with its order:



i. Prove that an AVL tree of order h contains as at most $2^{h+1} - 1$ nodes.

ii. Prove that an AVL tree of order h contains at least $F_{h+3} - 1$ nodes, where F_{h+3} is the (h+3)rd Fibonacci number.

Your result from (ii) can be used to show that AVL trees have a height that is roughly logarithmic in the number of nodes. There is a remarkable result about the Fibonacci numbers called *Binet's Formula*, which states that

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

Because the value of $\frac{-1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$ is never any less than -1, we get that

$$F_n \geq \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - 1$$

The number $\frac{1+\sqrt{5}}{2}$ is called the *golden ratio* and is typically denoted φ. This gives us that

$$F_n \geq \frac{\varphi^n}{\sqrt{5}} - 1$$

Using your result from part (ii), we have that in an AVL tree of n nodes, the height h of the tree is (approximately) related to n as follows:

$$n \geq \frac{\varphi^{h+3}}{\sqrt{5}} - 2$$

Simplifying, we get that $\log_{\varphi}((n+2)\sqrt{5}) - 3 \geq h$. In other words, the height of the tree is no greater than some logarithmic function of the number of nodes. This means that the time required to look up a node in the tree by following some path from the root downward will never be very large relative to the number of nodes in the tree.