Homework Assignment #1

Due Time/Date

2:20PM Tuesday, September 10, 2024. Late submission will be penalized by 20% for each working day overdue. Those who enroll late may be allowed an extension upon request.

How to Submit

Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the instructor's desk before the class on the due date starts. For early or late submissions, please drop them in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

There are five problems in this assignment, each accounting for 20 points. You must use *induction* for all proofs. (Note: problems marked with "(X.XX)" are taken from [Manber 1989] with probable adaptation.)

1. Consider *proper* binary trees, where every internal (non-leaf) node has two children. For any such tree T, let l_T denote the number of its leaves and m_T the number of its internal nodes. Prove *by induction* that $l_T = m_T + 1$.

2. Prove *by induction* that every natural number greater than or equal to 12 is a non-negative linear combination of 4 and 5, i.e., for every $n \in \mathbb{N}$, if $n \geq 12$, then there exist $a, b \in \mathbb{N}$ s.t. $n = 4a + 5b$ (where $\mathbb{N}$ is the set of all natural numbers, including 0).

3. Let $a_1, a_2, \cdots, a_n$ be positive real numbers such that $a_1 a_2 \cdots a_n = 1$. Prove *by induction* that $(1 + a_1)(1 + a_2) \cdots (1 + a_n) \geq 2^n$. (Hint: in the inductive step, try introducing a new variable that replaces two chosen numbers from the sequence.)

4. (2.37) Consider the recurrence relation for Fibonacci numbers $F(n) = F(n-1) + F(n-2)$. Without solving this recurrence, compare $F(n)$ to $G(n)$ defined by the recurrence $G(n) = G(n-1) + G(n-2) + 1$. It seems obvious that $G(n) > F(n)$ (because of the extra 1). Yet the following is a seemingly valid proof (by induction) that $G(n) = F(n) - 1$. We assume, by induction, that $G(k) = F(k) - 1$ for all k such that $1 \leq k \leq n$, and we consider $G(n+1)$:

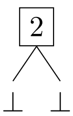
$$G(n+1) = G(n) + G(n-1) + 1 = F(n) - 1 + F(n-1) - 1 + 1 = F(n+1) - 1$$

What is wrong with this proof?

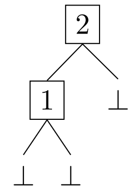
5. The set of all binary trees that store non-negative integer key values may be defined inductively as follows.

   - The empty tree, denoted $\perp$, is a binary tree, storing no key value.
   - If t_l and t_r are binary trees, then $node(k, t_l, t_r)$, where $k \in \mathbb{Z}$ and $k \geq 0$, is a also binary tree with the root storing key value k.

So, for instance, $node(2, \perp, \perp)$ is a single-node binary tree storing key value 2 and $node(2, node(1, \perp, \perp), \perp)$ is a binary tree with two nodes — the root and its left child, storing key values 2 and 1, repsectively. Pictorially, they may be depicted as below.



$node(2, \perp, \perp)$

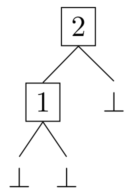


$node(2, node(1, \perp, \perp), \perp)$

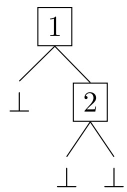
(a) (10 points) Define inductively a function Max that determines the largest of all key values of a binary tree. Let $Max(\perp) = 0$, though the empty tree does not store any key value. (Note: use the usual mathematical notations; do not write a computer program.)

(b) (5 points) Suppose, to differentiate the empty tree from a non-empty tree whose largest key value happens to be 0, we require that $Max(\perp) = -1$. Give another definition for Max that meets this requirement; again, induction should be used somewhere in the definition.

(c) (5 points) Consider counting the number of binary trees of different shapes, ignoring the key values they store.

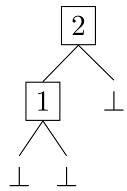


$node(2, node(1, \perp, \perp), \perp)$

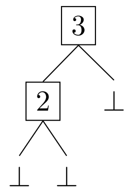


$node(1, \perp, node(2, \perp, \perp))$

The above two trees have different shapes, though they store the same key values.



$node(2, node(1, \perp, \perp), \perp)$



$node(3, node(2, \perp, \perp), \perp)$

The above two trees have the same shape, though they store different key values.

Let $b(n)$ denote the total number of distinctively-shaped binary trees with n nodes; for example, $b(0) = 1$, $b(1) = 1$, $b(2) = 2$, and $b(3) = 5$. Write a recurrence relation that defines $b(n)$, for $n \geq 0$.