Homework Assignment #2

Due Time/Date

1:20PM Tuesday, September 24, 2024. Late submission will be penalized by 20% for each working day overdue.

How to Submit

Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the table in front of 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 again the inductive definition in HW#1 for the set of all binary trees that store non-negative integer key values:

   - 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 also a binary tree with the root storing key value k.

   (a) Refine the definition to include only binary *search* trees where an inorder traversal of a binary search tree produces a list of all stored key values in *increasing* order.

   (b) Further refine the definition to include only AVL trees, which are binary search trees where the heights of the left and the right children of every internal node differ by at most 1. You may reuse the function discussed in class for computing the height of a given binary tree.

   In each case, the new definition should remain inductive.

2. Prove *by induction* that the regions formed by a planar graph all of whose vertices have even degrees can be colored with two colors such that no two adjacent regions (sharing one or more edges) have the same color.

3. (2.30) A **full binary tree** is defined inductively as follows. (Note: some authors prefer using the name "perfect binary tree" or "complete binary tree", while reserving "full binary tree" for another variant of binary trees.) A full binary tree of height 0 consists of 1 node which is the root. A full binary tree of height $h + 1$ consists of two full binary trees of height h whose roots are connected to a new root. Let T be a full binary tree of height h. The **height** of a node in T is h minus the node's distance from the root (e.g., the root has height h, whereas a leaf has height 0). Prove *by induction* that the sum of the heights of all the nodes in T is $2^{h+1} - h - 2$.

4. Consider the following two-player game: given a positive integer N, player A and player B take turns counting to N. In her/his turn, a player may advance the count by 1 or 2. For example, player A may start by saying "1, 2", player B follows by saying "3", player A follows by saying "4", etc. The player who eventually has to say the number N loses the game.

   A game is *determined* if one of the two players always has a way to win the game. Prove *by induction* that the counting game as described is determined for any positive integer N; the winner may differ for different given integers. (Hint: think about the remainder of the number N divided by 3.)

5. Consider the following algorithm for computing the square of the input number n, which is assumed to be a positive integer.

   Algorithm mySquare(n);
   begin
   // assume that $n > 0$
   $x := n$;
   $y := 0$;
   while $x > 0$ **do**
   $y := y + 2 \times x - 1$;
   $x := x - 1$
   od;
   mySquare := y
   end

   State a suitable loop invariant for the while loop and prove its correctness *by induction*. The loop invariant should be strong enough for deducing that, when the while loop terminates, the value of y equals the square of n.