Homework Assignment #2

Due Time/Date

2:20PM Tuesday, September 20, 2022. Late submission will be penalized by 20% for each working day overdue.

How to Submit

Please use a word processor or scan hand-written answers to produce a single PDF file. Name your file according to this pattern: "b107050xx-hw2". Upload the PDF file to the NTU COOL site for Algorithms 2022. 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 a also 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.

2. Reprove the following theorem which we have proven (mostly) in class. This time you must apply the *reversed induction* principle, or a variant of it, in some part of the proof. You may reuse some of the results that we have obtained in class without giving detailed proofs. Your main task is to demonstrate the use of reversed induction.

   > There exist Gray codes of length $\lceil \log_2 k \rceil$ for any positive integer $k \geq 2$. The Gray codes for the *even* values of k are *closed*, and the Gray codes for *odd* values of k are *open*.

3. (2.30) A **full binary tree** is defined inductively as follows. 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 that the sum of the heights of all the nodes in T is $2^{h+1} - h - 2$.

4. (2.23) The **lattice** points in the plane are the points with integer coordinates. Let P be a polygon that does not cross itself (such a polygon is called **simple**) such that all of its vertices are lattice points (see Figure 1). Let p be the number of lattice points that are on the boundary of the polygon (including its vertices), and let q be the number of lattice points that are inside the polygon. Prove that the area of polygon is $\frac{p}{2} + q - 1$.

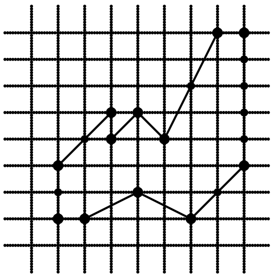

Figure 1: A simple polygon on the lattice points.

5. Consider the following pseudocode that represents the selection sort. The elements of an array of size n are indexed from 1 through n. Function *indexofLargest* gives the index of the largest element of the input array within the specified range of indices.

Algorithm selectionSort(A, n);
begin
    // the number of elements in A equals $n > 0$
    $last := n$;
    while $last > 1$ **do**
        $m := indexofLargest(A, 1, last)$;
        $A[m], A[last] := A[last], A[m]$; // swap
        $last := last - 1$;
    od;
end

State a suitable loop invariant for the main loop and prove its correctness.