Homework Assignment #2

Due Time/Date

2:10PM Tuesday, September 29, 2020. Late submission will be penalized by 20% for each working day overdue.

Note

Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time 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:

   (a) The empty tree, denoted $\perp$, is a binary tree.

   (b) If t_l and t_r are binary trees, then $node(k, t_l, t_r)$, where $k \in Z$ and $k \geq 0$, is also a binary tree.

   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. Then, define inductively a function that outputs the rank of a given key value (the position of the key value in the aforementioned sorted list, starting from position 1) if it is stored in the tree, or 0 if the key is not in the tree.

2. Consider the following recurrence relation:

$$\begin{cases} T(0) = 0 \\ T(1) = 1 \\ T(h) = T(h-1) + T(h-2) + 1, \quad h \geq 2 \end{cases}$$

   Prove by induction the relation $T(h) = F(h+2) - 1$, where $F(n)$ is the n-th Fibonacci number ($F(1) = 1$, $F(2) = 1$, and $F(n) = F(n-1) + F(n-2)$, for $n \geq 3$).

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 $\frac{p}{2} + q - 1$.

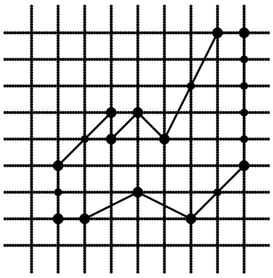

Figure 1: A simple polygon on the lattice points.

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.