Homework Assignment #2

Note

This assignment is due 2:10PM Monday, March 15, 2010. 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 II. Late submission will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

There are five problems in this assignment, each accounting for 20 points. You must use *induction* for all proofs. There is also a bonus problem, which is worth 20 points.

Problems

1. (2.24) We can define **anti-Gray codes** in the following way. Instead of minimizing the difference between two consecutive strings, we can try to maximize it. Is it possible to design an encoding for any even number of objects such that each pair of two consecutive strings differ by k bits (where k is the number of bits in each string)? How about $k-1$ bits (or $k-2$, $k-3$, etc.)? If it is possible, find an efficient construction.

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. (2.39) Design an algorithm to convert a binary number to a decimal number. The algorithm should be the opposite of algorithm *Convert_to_Binary* (see Fig. 1). The input is an array of bits b of length k, and the output is a number n. Prove the correctness of your algorithm by using a loop invariant.

Algorithm Convert_to_Binary(n):
<u>**Input:**</u> n (a positive integer).
<u>**Output:**</u> b (an array of bits corresponding to the binary representation of n).

```
begin
    $t := n$ ;
    $k := 0$ ;
    while $t > 0$ do
        $k := k + 1$ ;
        $b[k] := t \; mod \; 2$ ;
        $t := t \; div \; 2$ ;
end
```

Figure 1: Algorithm *Convert_to_Binary.*

5. (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 Fig. 2). 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 $p/2 + q - 1$.

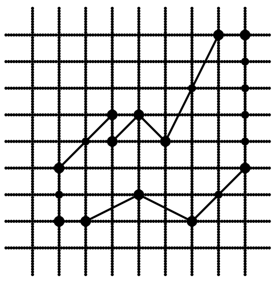

Figure 2: A simple polygon on the lattice points.

6. (Bonus) 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.

   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.*