CS/Math 240: Intro to Discrete Math	2/17/2011

Homework 4

Instructor: Dieter van Melkebeek

Guidelines:

This assignment covers structural induction, invariants, and program correctness. It is due on 3/1 at the beginning of class. Good luck!

Questions:

1. Consider the following inductive definition of a good sequence of 0's and 1's.

   Foundation The empty sequence is a good sequence.

   Constructor If s and t are good sequences, then so are $0s1t$ and $1s0t$.

   Show that

   (a) every good sequence consists of an equal number of 0's and 1's, and

   (b) every sequence consisting of an equal number of 0's and 1's is good.

2. In class we showed that the 8-puzzle is unsolvable. In this problem you'll show that the same holds for the 16-puzzle.

   Recall that in the 16-puzzle, there are 15 tiles, numbered 1 through 15, and one empty square arrange in a 4×4 grid. Any numbered tile adjacent to empty square can be slid into the empty square. The goal is to realize the following transformation:

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

$\rightarrow$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

   (a) By considering the number of pairs of positions that are in error, show that every solution requires an odd number of column moves.

   (b) By considering the position of the empty square, show that every solution requires an even number of column moves.

   (c) Conclude that no solution can exist.

3. Consider the following program specification:

   Input: An integer $n \geq 0$, an array $A[0..n-1]$ of integers, and an integer x.
   Output: -1 if x does not appear in A, otherwise the smallest index i such that $A[i] = x$.

   Consider the following implementation:

```
LinearSearch(n, A, x)
(1)   i ← 0
(2)   while i < n
(3)       if A[i] = x then return i
(4)       i ← i + 1
(5)   return -1
```

   (a) Establish the following loop invariant right before the while test in line (2) is executed: $0 \leq i \leq n$ and x does not appear in $A[0..(i-1)]$ (with the convention that $A[0..-1]$ denotes an empty array).
   (b) Prove the correctness of the implementation by arguing partial correctness and termination.

4. Consider the following program:

```
GCD(a, b)
(1)   if a ≤ b then (x, y) ← (a, b)
(2)            else (x, y) ← (b, a)
(3)   while x > 0
(4)       r ← y − ⌊y/x⌋x
(5)       y ← x
(6)       x ← r
(7)   return y
```

   (a) Show that the program correctly computes $\gcd(a, b)$ for any two natural numbers a and b, at least one of which is nonzero. Note that the latter condition is needed for the gcd to be well-defined, as every integer is a divisor of zero so $\gcd(0, 0)$ is not well-defined.
   (b) Show that the following invariant holds right before the while test in line (3) is executed: There exist integers w, z, u, and v such that $x = w \cdot a + z \cdot b$ and $y = u \cdot a + v \cdot b$.
   (c) Extend the algorithm such that is also outputs integers u and v such that $\gcd(a, b) = u \cdot a + v \cdot b$.
   Note: The mere existence of such integers is a nontrivial fact.

5. Recall the postage example from class, where we showed that with an unlimited supply of stamps of $a = 4$ cents and $b = 7$ cents, we can exactly realize every integer cent amount above a certain finite threshold. Show that for any pair (a, b) of positive integers, the latter property holds iff $\gcd(a, b) = 1$.
   Hint: Use the observation from part 4(c).