COT 5407 Introduction to Algorithms

Homework 3

DUE: Tuesday, October 5, 2010

Please remember that all submissions are to be your own work and must be typeset. Handwritten submissions will NOT be accepted.

1. A singly-linked list is *well-formed* if it is empty or if when following pointers from the first node, null is reached before any node is repeated; a repeated node would indicate that there is a cycle of pointers. Describe an algorithm that tests for cyclicity in time linear in the number of nodes reachable from the first node, without changing any pointer and using only constant extra space (a constant number of pointer variables). Note that you do not know the number of nodes that are in the list, and any algorithm that marks each node as it is visited will use too much extra space.

2. **(Dan Sleator)** There is a prize hidden in a box; the value of the prize is a positive integer N, which you don't know. To win the prize, you have to guess N. Your goal is to do it in as few guesses as possible. You start with a number of chips (specified below). Each chip allows you one guess that's too high. If you guess too high, and you have no chips, you lose. So, for example, if you start with no chips, then you can win in N guesses by simply guessing the sequence 1, 2, 3, ..., N.

   Suppose you start with 1 chip. Describe a strategy that makes $o(N)$ guesses. Find a function $g_1(N) = o(N)$ which is an upper bound on the number of guesses your strategy needs.

3. **(Kevin Wayne)** Find 2^N strings, each of length $2N$ that have the same `hashCode()` value, assuming that `String.hashCode()` follows the Java standard:

```
public int hashCode()
{
   int hash = 0;

   for( int i = 0; i < length( ); i++ )
      hash = hash * 31 + charAt( i );

   return hash;
}
```

4. Prove that given two binary search trees with the same set of keys, you can always transform one into the other by a sequence of left and right rotations.

5. Prove that a binary heap for eight items can be constructed in eight comparisons.