CSE326 Homework 1

Due: Monday, June 27

1. Weiss, 1.8a, 1.8b, 1.12a.

2. Weiss, 2.1.

3. Prove that for any constant k, $\log^k N = o(N)$. This can be proved either by showing appropriate settings of the constants from the definition of o, or from L'Hopital's rule.

4. Consider the following algorithm for counting the number of 1s in the binary representation of an integer:

```
countOnes( integer x )
    if x = 0
       return 0
    else
      return (x mod 2) + countOnes( x / 2 )
```

   Prove using induction that `countOnes` correctly returns the number of 1s in the binary representation of x.

5. The classic way to evaluate a polynomial is Horner's Rule. Horner's rule can be stated recursively as follows: Let $p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_kx^k$ be a polynomial. To evaluate the polynomial p at c, first let r be the result of evaluating the polynomial $a_1 + a_2x + a_3x^2 + \ldots + a_kx^{k-1}$ at c, then evaluate the final result as $a_0 + cr$.

   (a) Give pseudocode for a recursive function implementing Horner's Rule. Assume that the coefficients of the polynomial are stored in an array A with $A[i] = a_i, i = 0 \ldots k$. You will need to think about the parameters of your function carefully. Keep in mind that the subarray $A[i \ldots k]$ can be thought of as a polyomial of degree $k - i$.

   (b) Give pseudocode for an iterative algorithm (no recursion) for Horner's Rule.