CS3933 Winter 2007 – Solo Assignment 1
Due Wednesday January 31, 2007, at 5pm.

1. List the following functions in ascending order of growth rate.

   - $n^{\log_2 n}$
   - $n^2 \log_2 n$
   - $4^{\log_2 n}$
   - $\sqrt{n}$
   - 2^{2n}
   - $\sqrt{n}^5$
   - $\sqrt{n}^n$

2. Prove or disprove, using the definitions of O, Ω, and Θ, that

   (a) $3n \log_2 n \ \in \ O(5n^2)$

   (b) $2^n \ \notin \ \Omega(4^n)$

3. Consider the following algorithm:
   Algorithm *Spin*(n):
   Input: positive integer n.
   $x \leftarrow 0$
   for i from 1 to n
     for j from i to n
       $x \leftarrow x + 1$
   return x

   Analyse the running time of this algorithm in terms of the size of its input (rather than in terms of n, the input value itself). Use O, Ω, and Θ, as appropriate.

4. Consider the following algorithm:

Algorithm *Run*(A, n, m):
Input: 2-dimensional array A of real numbers, with n rows and m columns.

```
i ← 0
j ← 0
while i < n − 1 or j < m − 1 do
    while (A[i, j] < A[i + 1, j] and i < n − 1)
        i ← i + 1
        print "down"
    j ← j + 1
    while (A[i, j] < A[i, j + 1] and j < m − 1)
        j ← j + 1
        print "right"
    i ← i + 1
```

Analyse the running time of the algorithm using O, Ω, and Θ, as appropriate.

5. Consider the following algorithm:

Algorithm *Numbers*(A,n):
Input: array A of n positive integers.

```
S ← 0
for i from 0 to n − 1 do
    for j from 0 to i do
        P ← 1
        for k from i to n − 1 do
            P ← P ∗ A[k]
        for k from i to j do
            P ← P ∗ A[k]
        S ← S + P
return S
```

(a) Analyse the running time of this algorithm, using O, Ω, and Θ, as appropriate.

(b) Now that you have determined the degree of the polynomial, figure out the exact polynomial function for how many times the statement $P \leftarrow P * A[k]$ is executed in terms of the input size n.

Determine the polynomial by implementing the algorithm, testing it for various

sizes n, and solving for the unknown polynomial coefficients. Please submit your test case results (n and corresponding count) as well as your polynomial function.

Note that how you solve for your coefficients is not an essential part of the assignment; it's just a step in finding the function (so you do not need to show your work for the coefficient solving, and indeed can solve them however you like).

6. Consider a modified mergesort algorithm so that it splits the input not into two sets of almost-equal sizes, but into three sets of sizes approximately one-third. Write an algorithm for this ternary sorting algorithm. Asymptotically analyze it by establishing a recurrence equation first, then analysing the recurrence tree of the algorithm. You can make assumptions about the instance sizes considered in order to simplify your math, as long as the sizes can still be arbitrarily large.