Algorithms — Homework 2
Runtime Analysis

Due: September 15.

1) Rank the following functions by order of growth. That is, find an arrangement $f_1, f_2, \ldots$ of the functions satisfying $f_1 \in \mathcal{O}(f_2)$, $f_2 \in \mathcal{O}(f_3)$, $\ldots$. Partition your list into equivalence classes such that functions f_i and f_j are in the same class if and only if $f_i \in \Theta(f_j)$.

$$\left(\sqrt{2}\right)^{\log n} \quad n^2 \quad n! \quad \left(\tfrac{3}{2}\right)^n \quad \log^2 n$$

$$4^{\log n} \quad n \quad 2^n \quad n \log n \quad 2^{2^{n+1}}$$

2) Find two functions $f(n)$ and $g(n)$ that satisfy the following relationship. If no such f and g exist, shortly explain why.

   a) $f(n) \in o(g(n))$ and $f(n) \notin \Theta(g(n))$
   b) $f(n) \in \Theta(g(n))$ and $f(n) \in o(g(n))$
   c) $f(n) \in \Theta(g(n))$ and $f(n) \notin \mathcal{O}(g(n))$
   d) $f(n) \in \Omega(g(n))$ and $f(n) \notin \mathcal{O}(g(n))$

3) Use a recursion tree to determine a good asymptotic upper bound on the recurrence T. Use the master theorem for a) and the substitution method for b) to verify your answer.

   a) $T(n) = 3T(n/2) + n$
   b) $T(n) = 2T(n-1) + 1$