Foundations of Algorithms
Homework 1

Arthur Nunes-Harwitt

1. Rank the following functions by order of growth. Further, partition the list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) \in \Theta(g(n))$. (See page 58 in CLRS for a definition of $\lg^*(n)$.)

$\ln(\ln(n))$	$\lg^*(n)$	$n2^n$	$n^{\lg(lg(n))}$	$\ln(n)$	1
$2^{\lg(n)}$	$(\lg(n))^{\lg(n)}$	e^n	$4^{\lg(n)}$	$(n+1)!$	$\sqrt{\lg(n)}$
$(\frac{3}{2})^n$	n^3	$(\lg(n))^2$	$\lg(n!)$	2^{2^n}	$n^{1/\lg(n)}$
$\lg^*(\lg(n))$	$2^{\sqrt{2\lg(n)}}$	n	2^n	$n\lg(n)$	$2^{2^{n+1}}$
$\lg(\lg^*(n))$	$2^{\lg^*(n)}$	$(\sqrt{2})^{\lg(n)}$	n^2	$n!$	$(\lg(n))!$

2. Rank the following functions of x by order of growth. Note: the constants a,b,c,k are all greater than one.

   $\sqrt[k]{x}$, a^x, x^c, $\log_b(x)$

3. (a) Using the class definition of $\mathcal{O}$, prove that $n = \mathcal{O}(n^2)$.
   (b) Using the class definition of $\mathcal{O}$, prove that $n^2 = \mathcal{O}(n^2)$.
   (c) Using the class definition of $\mathcal{O}$, prove that $3n^2 + 5n = \mathcal{O}(n^2)$.

4. (a) Using the class definition of $\mathcal{O}$, prove that $n^k = \mathcal{O}(n^{k'})$ if $k \leq k'$.
   (b) Using the class definition of $\mathcal{O}$, prove that $\mathcal{O}(f(n)) + \mathcal{O}(f(n)) = \mathcal{O}(f(n))$.

5. (a) Given that $\displaystyle\sum_{k=2}^{n} \frac{1}{k} \leq \ln(n) - \ln(1)$, using the class definition of $\mathcal{O}$, prove that $H_n \in \mathcal{O}(\ln(n))$.
   (b) Given that $\displaystyle\sum_{k=2}^{n} \frac{1}{k} \geq \ln(n+1) - \ln(2)$, using the class definition of Ω, prove that $H_n \in \Omega(\ln(n))$.

6. (**project**) Recall the definition of the Fibonacci numbers.

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{aligned}$$

   Write a recursive function `fib` that implements the above recurrence. What is the smallest n such that you notice `fib` running slowly?

7. Consider the following recurrence.

$$\begin{array}{rcl} f(0;a,b) & = & a \\ f(1;a,b) & = & b \\ f(n;a,b) & = & f(n-1;b,a+b) \end{array}$$

   (a) Prove using mathematical induction that for any $n \in \mathbb{N}$ if $n > 1$ then $f(n;a,b) = f(n-1;a,b) + f(n-2;a,b)$.

   (b) Prove using the strong form of mathematical induction that for any $n \in \mathbb{N}$, $F_n = f(n;0,1)$. You should use the previous result in your proof.

8. (**project**) Write a recursive function `fibItHelper` that takes three arguments, n, a, and b; it should implement the recurrence f. Then write a function `fibIt` that calls `fibItHelper` initializing a to 0 and b to 1. Does `fibIt` also run slowly on the value of n that you found made `fib` run slowly?