Emily Crilley

Analysis of Algorithms - Homework 1

1.

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

2. $log_b(x) < \sqrt[k]{x} < x^c < a^x$

3.

a. $n = O(n^2)$ $k = 1$

$\frac{n}{n^2} \leq \frac{Cn^2}{n^2} = 1$

n $\leq n^2$, meaning n = O(n^2)

b. $n^2 = O(n^2)$

$\frac{n^2}{n^2} \leq \frac{Cn^2}{n^2}$, C = 1

$\frac{n^2}{n^2} = \frac{n^2}{n^2}$

c. $3n^2 + 5n = O(n^2)$

$\frac{3n^2 + 5n}{n^2} \leq \frac{3n^2 + 5n^2}{n^2}$, C = 8

$|3n^2 + 5n| \leq 8 * |n^2|$

4.

a. Assume $n \geq 1,\ c = 2$

$n \geq 1$ *which becomes* $n^2 > n$

$ln(n^2) \geq ln(n)$

$2\,ln(n) \geq ln(n)$

$2ln(n) - ln(1) \geq ln(n) - ln(1)$

With Given Information, $\sum_{k=2}^{n} 1/k \leq 2ln(n)$

$|\sum_{k=2}^{n} 1/k| \leq |2ln(n)|$

b. Assume $n \geq 1,\ c = 2$

$n - 1 \geq 0$

$(n-1)^2 \geq 0^{\,2}$

$(n-1)(n-1) \geq 0$

$(n^2 - 2n + 1) \geq 0$

$(n^2 - 2n + 1) \geq 4n$

$\frac{n^{2-2n+1}}{4} \geq n$

$(\frac{n+1}{2})^2 \geq n$

$ln((\frac{n+1}{2})^2) \geq ln(n)$

$ln(\frac{n+1}{2}) \geq 1/2ln(n)$

$ln(n+1) - ln(2) \geq 1/2ln(n)$

With Given Information, $\sum_{k=2}^{n} 1/k \ \geq 1/2ln(n)$

$|\sum_{k=2}^{n} 1/k| \ \geq \ |.5ln(n)|$

5. See hw1.py for code. I ran my fib function between the numbers 1-1000 and I started to notice it ran slow around where i = 33. My times jumped from .76 to 1.17, it steadily increases from here, getting numbers like 60.74 where i = 41
6.
   a. Proof using best case, n = 2

      $f(2; a, b) \ = f(2-1; a, b) \ + \ f(2-2; a, b)$

      $f(2-1; a, b) \ + \ f(2-2; a, b) \ = \ f(1; a, b) + f(0; a, b)$

      = b + a
   b. $\sum_{i=1}^{n-1} T(i) + T(0)$, where T(i) = b+a

      $\sum_{i=1}^{n-1}(a \ + \ b) + T(0)$

      $\sum_{i=1}^{n-1}(a \ + \ b) \ + \ 1$

      $(a \ + \ b \ - \ 1) \ + \ 1$

      = a + b
7. See hw1.py for code. I found that it runs much faster at the same values. It rarely took more than 0.0 time to compute.