Log Worksheet

1. If $\log_{100}x = y$, express $\log_{10}x^3$ in terms of y?

$$3\, log_{10}\, x = 6y$$

$$log_{10}\, x = \frac{log_{100}\, x}{log_{100}\, 10}$$

$$= \frac{y}{1\,/\,2} = 2y$$

2. Prove that $\log(n!) = O(n\log n)$.

$$n! \le n^n, n \ge 1$$

$$n \times (n-1) \times (n-2) \times \ldots \times 1 \le n \times n \times n \ldots \times n$$

$$log\, n! \le log\, n^n$$

$$n\, log\, n, n \ge 1$$

$$c = 1$$

3. Prove that $\log(n!) = \Omega(n\log n)$ (difficult).

$$n! \ge (\frac{n}{2})^{(n/2)}$$

$$log\, n! \ge \frac{n}{2} log \frac{n}{2}$$

$\ge c\, n\, log\, n$ → want to find constant c

$(1-c)n\, log\, n \ge (log\, e)n$ → must satisfy n ≥ n_0

$(1-c)n\, log\, n \ge (log\, e)$ → must satisfy n > 0

$(1-c)logn \ge (1-c)logn_0$ → need positive c and n_0 to satisfy the statement below

$$(1-c)logn_0 \ge (log\, e)$$

For example, c = 0.1 and n_0 = 4

$$log\, n! = \theta(n\, log\, n)$$

Worst/Best/Average vs $O/\Omega/\Theta$

Students often confuse the concepts of worst case, best case and average case analysis with the three kinds of bounds (O,Ω,Θ). The purpose of this exercise is to understand the interplay between these two concepts.

Suppose an algorithm takes 50, 60, 70, 80, and 100 units of time on inputs A, B, C, D and E, respectively (and suppose these are the only inputs possible for the problem)

1. What is the best case time of the algorithm and on which input?

   Input A - 50

2. What is the worst case time of the algorithm and on which input?

   Input E - 100

3. What is the average case time of the algorithm and on which input (trick question)?

   finding average = 50+60+70+80+100/5 = 72 – no input

4. For the best case time (from item 1 above), provide an integer that

   (a) serves as an upper bound (O analogy),

   60

   (b) serves as a lower bound (Ω analogy)

   40

   (c) *simultaneously* serves as an upper and lower bound (Θ analogy). equal

   50

5. Why is this example an “analogy” and not the real thing?

   This is an analogy since this algorithm is using the relationships between pairs of related items to learn about relationships between a “real thing”. A “real thing” would have more code examples, rather than a simple input output algorithm.

6. Repeat for the other two cases; i.e., worst (from item 2) and average (from item 3).

   Worst case
     Upper bound 110
     Lower bound 90
     Simultaneously 100
   Average vase
     Upper bound 80
     Lower bound 60
     Simultaneously 72

Practical Analysis

Assume array *A* is indexed from 1 to *n*.

INEFFICIENT SORT(*A*, *n*)

for *i* = 1 **to** *n*! **do**

Boolean *sortedSoFar* = TRUE;

j = 1;

P = nextPermutation(*A*); // theta(n)

while *j* < *n* **and** *sortedSoFar* **do**

if *P*[*j*] > *P*[*j* + 1]

then *sortedSoFar* = FALSE // O(n)

j++

if (*sortedSoFar*) **then output** *P* //theta(n)

Analyze the worst-case complexity of INEFFICIENT SORT assuming that the nextPermutation function always takes $\Theta(n)$ time.

Complexity: theta(n. n!) = theta(n!)

lim(n!/(n n!) = lim(1/n) = 0

Your answer should fit above the line!