HW#1

Due date: 3/21

1 This problem asks you to profile various versions merge sort and insertion sort. For each version, you shall turn in the source code and profiling data. You shall profile on at least two input sizes, each taking at least 3 runs. The input sizes to profile are up to you.

You shall indicate the platform (i.e. machine and OS) and the compiler used for profiling.

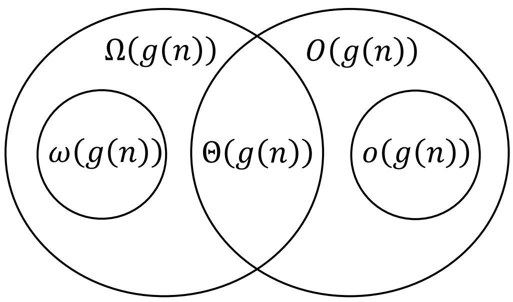
a) Profile the ordinary merge sort as given in the book and lecture (pp.24~25). Profile the refined merge sort as given in the lecture (p.31).
Use your profiling data to figure out the percentage of time the refined merge sort saves against the ordinary merge sort. (20%)

b) Profile the ordinary insertion sort as given in the book and lecture (p.19). Profile the refined insertion sort as given in the lecture (p.32).
Use your profiling data to figure out the percentage of time the refined insertion sort saves against the ordinary insertion sort. (20%)

c) Use your profiling data of part a) and b) to determine the threshold for the refined merge sort to beat the refined insertion sort.
Modify the refined merge sort by turning it to the refined insertion sort when the array size is less than the threshold value.
Profile this mixed-up merge sort.
Use your profiling data to figure out the percentage of time the mixed-up merge sort saves against the refined merge sort. (20%)

2 Do Problem 2-1 a) 5%, b) 10%, c) 10%, d) 5%)

3 In this problem, we shall justify the following diagram:



First of all, $o(g(n)) \subseteq O(g(n))$ and $\omega(g(n)) \subseteq \Omega(g(n))$ are trivial.

a) Do Exercise 3.1-5, i.e. prove Theorem 3.1, which states that (10%)
$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

b) Show that $o(g(n)) \cap \Omega(g(n)) = \emptyset$, whence (10%)
$o(g(n)) \subseteq O(g(n)) - \Theta(g(n))$

c) Give an example to show that $o(g(n)) \neq O(g(n)) - \Theta(g(n))$ (5%)

d) Within the diagram, draw the sets (5%)

$$A = \left\{ f(n) : \lim_{n \to \to \infty} \frac{f(n)}{g(n)} = 0 \right\}$$

$$B = \left\{ f(n) : \lim_{n \to \to \infty} \frac{f(n)}{g(n)} = \infty \right\}$$

$$C = \left\{ f(n) : \lim_{n \to \to \infty} \frac{f(n)}{g(n)} = a, 0 < a < \infty \right\}$$

and explain.

Comment

The lower-bound counterparts of b) and c) are analogous and needn't be proven.

4 For each statement below, determine if it is *always true*, *never true*, or *sometimes true*. In case it is always true or never true, explain why. If it is sometimes true, give one example for which it is true, and one for which it is false. (25%)

a) Problem 3-4, d)

b) Problem 3-4, e)

c) Problem 3-4, h)

d) $o\big(f(n)\big) + o\big(f(n)\big) = o\big(f(n)\big)$

e) $f(n) = n^2 + O(n)$ and $g(n) = n^2 + O(n)$ implies $f(n) = g(n)$

5 True or false. You ***must*** justify your answers. [Past exam questions]

a) $O(n) = \{f(n) : \exists c > 0 \text{ such that } 0 \leq f(n) \leq cn \text{ for all } n > 0\}$ (5%)
Hint: Compare it with book's definition on big-O.
$O(n) = \{f(n) : \exists c > 0\ n_0 > 0 \text{ such that } 0 \leq f(n) \leq cn \text{ for all } n \geq n_0\}$

b) $O(n^k) = O(n)^k$ for any integer k (10%)
Hint: Consider $k < 0, k = 0,$ and $k > 0$

Readings

Visit the book's web site at http://mitpress.mit.edu/algorithms/ for solutions to selected exercises and problems.