HW#1 solution

1 Omitted

2 a) Insertion sort takes $\Theta(n^2)$

Each list has k elements, so that insertion sort takes $\Theta(k^2)$

There has $\frac{n}{k}$ lists, so totally takes $\Theta(\frac{n}{k} * k^2) = \Theta(nk)$

b) Pairwise merging starting with $\frac{n}{k}$ lists and finishing with 1 list

We can draw result lists as binary tree, so that tree's height is $\lg\frac{n}{k}$

Each list takes $\Theta(k)$ to put result, so that $\frac{n}{k}$ lists takes $\Theta(\frac{n}{k} * k) = \Theta(n)$

Finally, each level of tree takes $\Theta(n)$, so that it totally need $\Theta(n\lg\frac{n}{k})$

c) The modified algorithm takes $\Theta\left(nk + n\lg\frac{n}{k}\right)$

Obviously, k cannot more than $\Theta(\lg n)$

If k more than $\Theta(\lg n)$, modified algorithm will take more than $\Theta(n\,lg\,n)$

k = $\Theta(\lg n)$ into $\Theta\left(nk + n\lg\frac{n}{k}\right) = \Theta(nk + n\lg n - n\lg k)$

We can get $\Theta\big(n\lg n + n\lg n - n\lg\lg n\big) = \Theta\big(2n\lg n - n\lg\lg n\big)$

$= \Theta(n\lg n)$

d) The k should be the largest list length on which insertion sort is faster than merge sort

3 a) **LEMMA 1** $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

Proof of $\subseteq$

$f(n) = \Theta(g(n))$

$\Rightarrow \exists c_1, c_2 > 0\ and\ n_0$ such that $c_2 g(n) \le f(n) \le c_1 g(n)\ \ \forall n \ge n_0$

$\Rightarrow f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

Proof of $\supseteq$

$f(n) = O(g(n)) \Rightarrow \exists c_1 > 0\ and\ n_1$ such that $f(n) \le c_1 g(n)\ \ \forall n \ge n_1$

$f(n) = \Omega(g(n)) \Rightarrow \exists c_2 > 0\ and\ n_2$ such that $c_2 g(n) \le f(n)\ \ \forall n \ge n_2$

Thus,

$c_2 g(n) \le f(n) \le c_1 g(n)\ \ \forall n \ge \max(n_1, n_2)$

$\Rightarrow f(n) = \Theta(g(n))$

b) **LEMMA 2** $o(g(n)) \cap \Omega(g(n)) = \emptyset$,

Proof

$f(n) = \Omega(g(n))$

$\Rightarrow \exists c_1 > 0\ and\ n_1$ such that $c_1 g(n) \le f(n) \quad \forall n \ge n_1$ (1)

$f(n) = o(g(n))$

$\Rightarrow\ \forall c > 0\ \exists n_0$ such that $f(n) < cg(n)\ \ \forall n \ge n_0$

$\Rightarrow\ \exists n_0'$ (that depends on c_1) such that $f(n) < c_1 g(n)\ \ \forall n \ge n_0'$ (2)

It follows from (1) and (2) that

$c_1 g(n) \le f(n) < c_1 g(n) \quad \forall n \ge \max(n_1, n_0')$

which is impossible.

COROLLARY $o(g(n)) \subseteq O(g(n)) - \Theta(g(n))$

Proof

Let $f(n) = o(g(n))$

Then, $f(n) = O(g(n)) \qquad \because o(g(n)) = O(g(n))$

Also, $f(n) \ne \Omega(g(n)) \qquad \because$ LEMMA 2

Thus, $f(n) \ne \Theta(g(n)) \qquad \because$ LEMMA 1

Therefore, $f(n) = O(g(n)) - \Theta(g(n))$

c) Have to give an example to show that $O(g(n)) - \Theta(g(n)) \nsubseteq o(g(n))$

Let $f(n) = n(1 + \sin n)$, $g(n) = n$

Then, $f(n) = O\big(g(n)\big) \quad \because f(n) \le 2g(n) \quad \forall n \ge 0$

But, $f(n) \ne \Omega(g(n))$

$\because \nexists \mathrm{c} > 0$ such that $cg(n) \le f(n) = 0$, for $n = 2k\pi + 3\pi/2$, for any k

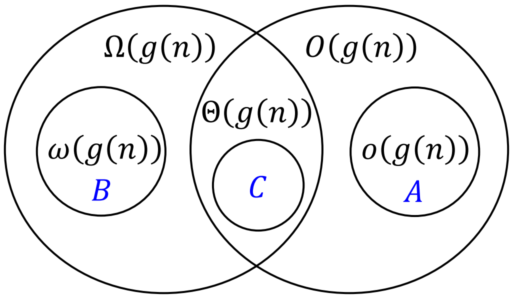
Thus, $f(n) = O(g(n)) - \Theta(g(n))$

But, $f(n) \ne o(g(n))\ \because \lim \frac{f(n)}{g(n)}$ doesn't exist.

Comment

For another example, see Chap 04, pp27~28.

d)



The three sets for A, B and C follow immediately from the theorems given in the lecture on Chap 03, pp13~15.

4 a) $f(n) = O(g(n)) \Rightarrow 2^{f(n)} = 2^{O(g(n))}$

Always true

This follows immediately from the definition of $2^{O(g(n))}$:

$2^{O(g(n))} = \{2^{f(n)} : f(n) = O(g(n))\}$

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

Sometimes true

For $f(n) = n$, it is true.

For $f(n) = 1/n$, it is false.

c) $f(n) + o(f(n)) = \Theta(f(n))$

Always true

Let $g(n) = o(f(n))$

Let $c > 0$ be any constant, then $g(n) < cf(n)$ for sufficiently large n

Thus, for sufficiently large n

$f(n) \le f(n) + g(n) \le (1+c)f(n)$

Another proof

Let $g(n) = o(f(n))$, then $\lim \frac{g(n)}{f(n)} = 0$

Thus, $\lim \frac{f(n)+g(n)}{f(n)} = 1 \Rightarrow f(n) + g(n) = \Theta(f(n))$

d) $O(f(n)) + O(f(n)) = O(f(n))$

Always true

$O(f(n)) + O(f(n)) = O(\max(f(n), f(n)) = O(f(n))$

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

Sometimes true

It could be

$f(n) = g(n) = n^2 + n.$

or

$f(n) = n^2 + 2n$, but $g(n) = n^2 + \lg n$

5 a) **Solution:** *True*

Let $\hat{O}(n) =$ book's definition on big-O.

We show that $O(n) = \hat{O}(n)$

$O(n) \subseteq \hat{O}(n)$

This is trivial – simply pick $n_0 = 1$.

$\hat{O}(n) \subseteq O(n)$

Let $f(n) = \hat{O}(n)$

Then, $0 \le f(n) \le cn \; \forall \, n \ge n_0$, for some c and n_0

For $n < n_0$, it is possible that $f(n) > cn$.

However, we may choose a large enough constant c' to handle these cases.

For each $0 < i < n_0$, let $c_i = f(i)/i$

Let $c' = \max\{c, \max_{0<i<n_0} c_i\}$

Then, $0 \le f(n) \le c'n \; \forall \, n > 0$

b) **Solution:** *False*

1 $O(n^k) = O(n)^k$ for $k > 0$

Proof of the one-way equality $O(n^k) = O(n)^k$

$f(n) = O(n^k)$

$\Rightarrow f(n) \le cn^k \quad \forall n \ge n_0$

$\Rightarrow f(n) \le \left(\sqrt[k]{c}n\right)^k \quad \forall n \ge n_0$

$\Rightarrow f(n) = O(n)^k \qquad \because O(n)^k = \{f^k(n) | f(n) = O(n)\}$

Proof of the one-way equality $O(n)^k = O(n^k)$

$f(n) = O(n)^k$

$\Rightarrow f(n) = g^k(n)$ for some $g(n) = O(n)$

$\Rightarrow f(n) \le (cn)^k \quad \forall n \ge n_0$

$\Rightarrow f(n) \le c^k n^k \quad \forall n \ge n_0$

$\Rightarrow f(n) = O(n^k)$

2 $O(n^0) \ne O(n)^0$

$\because O(n^0) = O(1)$.

But, $O(n)^0 = \{f^0(n) : f(n) = O(n)\}$

$= \{g(n) : g(n) = 1 \text{ for all large enough } n\}$

3 $O(n^k) \ne O(n)^k$ for $k < 0$.

In fact, $O(n)^k = \Omega(n^k)$ for $k < 0$. See Chap03 p.36.