Mục lục

Bài Tập 1:

1a) Ý nghĩa của độ phức tạp khi đề cập đến thuật toán:
- Mỗi bài toán có kích thước của đầu vào. Độ phức tạp là khái niệm tương đối để định lượng số phép toán của giải thuật so với kích thước của giá trị đầu vào.
- Gọi $T(n)$ là hàm số thể hiện độ phức tạp về thời gian, với n là kích thước đầu vào của bài toán. Độ phức tạp phân lớp theo cấp độ tăng của hàm $T(n)$ khi n đủ lớn.
- Giải thuật có độ phức tạp ở phân lớp thấp hơn thì có hiệu quả cao hơn.

b) Hãy cho biết ý kiến của bạn về nhận định dưới đây và giải thích vì sao?

"Khi nghiên cứu về các thuật toán, người ta quan tâm đặc biệt đến tính hiệu quả về thời gian của chúng nhưng thường là quan tâm đến bậc tăng trưởng (order of growth) của hàm thời gian thực hiện của thuật toán, chứ không phải là bản thân thời gian thực hiện T(n)".

Nhận định trên là đúng vì trên thực tế:

- Rất khó để tính chính xác T(n) và thời gian thực hiện thuật toán còn phụ thuộc vào nhiều yếu tố khác nhau (như yếu tố vật lý, trình biên dịch,…)
- Nếu tính được T(n) thì vẫn gặp khó khăn khi so sánh hàm số của các thuật toán

Vì vậy, ta cần thay thế bằng sự so sánh tương đối. Những hàm không sai biệt nhiều thì xem như sấp sỉ về độ lớn và chỉ quan tâm đến những giá trị n đủ lớn. Tức so sánh xem khi n tiến tới vô cùng thì hàm nào lớn hơn.
Tuy nhiên, phương pháp này có những hạn chế như thể hiện rất ít thông tin thực tế về thời gian chạy của chương trình hoặc thậm chí chỉ cho biết vấn đề có thể được giải quyết bằng thuật toán hay không.

c) Nói đến độ phức tạp, ta có 3 loại ký hiệu tiệm cận: O, Ω, θ

- Ký hiệu Big O: xác định tiêm cận trên của thuật toán (độ phức tạp trong trường hợp xấu nhất).

$$O(g(n) = \{ f(n): tồn\ tại\ các\ hằng\ số\ dương\ c\ và\ n_0\ sao\ cho\ 0 \le f(n) \le c * g(n)\ \forall n \ge n_0$$

- Ký hiệu Big Ω: xác định tiệm cận dưới của thuật toán (độ phức tạp trong trường hợp tốt nhất).

$$\Omega\big(g(n)\big) = \{f(n): tồn\ tại\ các\ hằng\ số\ dương\ c\ và\ n_0\ sao\ cho\ 0\ \le c * g(n) \le f(n)\ \forall n \ge n_0$$

- Ký hiệu Big θ: xác định cả tiệm cận trên và tiêm cận dưới của thuật toán.

$$\theta\big(g(n)\big) = \{ f(n): tồn\ tại\ các\ hằng\ số\ dương\ c_1, c_2\ và\ n_0\ sao\ cho\ 0 \le c_1 * g(n) \le f(n) \le c_2 * g(n)\ \forall n \ge n_0$$

Bài tập 2: Comparison of running times

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\log n$	2^{10^6}	2^{6*10^7}	2^{36*10^8}	2^{864*10^8}	$2^{25920*10^8}$	$2^{315360*10^8}$	$2^{31556736*10^8}$
$\sqrt{n}$	10^{12}	$36 * 10^{14}$	$1296 * 10^{16}$	$746496 * 10^{16}$	$6718464 * 10^{18}$	$994519296 * 10^{18}$	$995827586973696 * 10^{18}$
n	10^6	$6 * 10^7$	$36 * 10^8$	$864 * 10^8$	$2592 * 10^9$	$31536 * 10^9$	$31556736 * 10^8$
$n\,log\,n$	62746	2801417	133378058	2755147513	71870856404	797633893349	68654697441062

For

n^2	1000	7745	60000	293938	1609968	5615692	56175382
n^3	100	391	1532	4420	13736	31593	146677
2^n	19	25	31	36	41	44	51
n!	9	11	12	13	15	16	17

each function f(n) and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes f(n) microseconds.

Để tìm ra kích thước lớn nhất của vấn đề có thể được giải quyết kịp thời t, chúng ta cần giải phương trình sau để tìm n $f(n)= t$ in microseconds

*ví dụ:*kích thước lớn nhất mà vấn đề có độ phức tạp n! có thể tính trong 1 giây :

n! = $\frac{1}{\mu s} = \frac{1}{10^{-6}}$ =10^6=> n = 9 (nếu n =10 > 10^6)

1 phút: n! = $\frac{60}{\mu s} = \frac{60}{10^{-6}}$ =$60 * 10^6$ => n = 11

1 giờ : n! = $\frac{60}{\mu s} = \frac{60}{10^{-6}}$ =$3600 * 10^6$ => n = 12 (13! > $3600 * 10^6$)

….

Bài tập 3:

3a) phép suy diễn bên dưới là đúng hay sai vì sao ?

$$\frac{1}{2}n^2 = O(n^2)$$
$$n^2+1 = O(n^2)$$
$$\Rightarrow \frac{1}{2}n^2 = n^2+1 \quad ???$$

Phép biểu diễn trên là sai vì bản chất của biểu thức trên là :

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

dấu " = " chỉ là ký hiệu hình thức nên không thể suy diễn ra $\frac{1}{2}n^2 = n^2+1$

b) Xét: $f(n) = 7n^2$

$g(n) = n^2 - 80n$

$h(n) = n^3$

Chứng minh: $\boldsymbol{f(n) = O(g(n))}$

$\boldsymbol{g(n) = O(f(n))}$

$\boldsymbol{f(n) = O(h(n))}$

$\boldsymbol{h(n) \neq O(f(n))}$

$\boldsymbol{f(n) = O(g(n))}$

Chứng minh: $7n^2 \le c(n^2 - 80n) \quad \forall n \ge n_0$

Giả sử chọn $c = 8$ ta được:

$$n^2 - 640n \ge 0 \Rightarrow \left[\begin{matrix} n \le 0 \\ n \ge 640 \end{matrix}\right.$$

Vậy chọn $c = 8, n_0 = 640$

Theo định nghĩa của Big-O, ta được $f(n) = O(g(n))$ (đpcm)

$\boldsymbol{g(n) = O(f(n))}$

Ta thấy: $n^2 - 80n \le n^2 \le 7n^2 \ \forall n \ge 1$

Chọn $c = 7, n_0 = 1$

Theo định nghĩa của Big-O, ta được $g(n) = O(f(n))$ (đpcm)

$\boldsymbol{f(n) = O(h(n))}$

Ta thấy: $7n^2 \le 7n^3 \quad \forall n \ge 1$

Chọn $c = 7, n_0 = 1$

Theo định nghĩa của Big-O, ta được $f(n) = O(h(n))$ (đpcm)

$\boldsymbol{h(n) \neq O\big(f(n)\big)}$

Giả sử: $h(n) = O\big(f(n)\big)$

$\Rightarrow \exists c \in R^+, \exists n_0 \in N$ sao cho $n^3 \leq c(7n^2), \forall n \geq n_o$

Suy ra: $n^2(n-7c) \leq 0$

Xét: $n^2(n-7c)$

n	0	7c	
$n^2(n-7c)$	$-$	$-$	$+$

$\Rightarrow \nexists n_0 \rightarrow n^2(n-7c) \leq 0 \quad \forall n > n_0$

$\Rightarrow h(n) \neq O\big(f(n)\big)$ (đpcm)

3.c)

c) Chứng minh

$n^4 + n + 1 \notin O(n^2)$

$O(n^2) \neq O(n)$

$n \notin O(\log_2 n)$

- ***Giả thiết (1):*** $\boldsymbol{n^4 + n + 1 \notin O(n^2)}$

Giả sử: $n^4 + n + 1 \in O(n^2)$

$T(n) = n^4 + n + 1 \leq n^4 + n^4 + n^4, \forall n \geq 1$

$\Rightarrow T(n) \leq 3n^4 \; \forall n \geq 1$

Theo định nghĩa của Big–O, chọn c = 3 và n_0 = 1, ta có $T(n) = O(n^4)$ mà $O(n^4) \supset O(n^2)$, $O(n^4) \neq O(n^2)$

Vậy giả sử trên là sai. $T(n) \notin O(n^2)$ *hay* $n^4 + n + 1 \notin O(n^2) \Rightarrow$ *đccm*

- ***Giả thiết (2):*** $\boldsymbol{O(n^2) \neq O(n)}$

Giả sử $O(n^2) = O(n) \Rightarrow n \in O(n^2)$ và $n^2 \in O(n)$

Ta có $n^2 \in O(n^2)$ mà $O(n^2) \supset O(n), O(n^2) \neq O(n) \rightarrow n^2 \notin O(n)$

Vậy giả sử trên là sai $\Rightarrow O(n^2) \neq O(n) \Rightarrow$ đccm

- ***Giả thiết (3):*** $\boldsymbol{n \notin O(log_2 n)}$

Giả sử $n \in O(log_2 n)$

$T(n) = n < 2n \; \forall n \geq 1$

Theo định nghĩa của Big–O, chọn c = 2 và n_0 = 1, ta có $T(n) = O(n)$ mà $O(n) \supset O(log_2 n)$, $O(n) \neq O(log_2 n)$

Vậy giả thiết (3) là sai. $n \notin O(log_2 n) \Rightarrow$ *đccm*

Bài tập 4: (G1). Sắp xếp tăng dần theo bậc tăng trưởng Big – O

Group 1:
$f_1(n) = \binom{n}{100}$
$f_2(n) = n^{100}$
$f_3(n) = 1/n$
$f_4(n) = 10^{1000}n$
$f_5(n) = nlogn$

$$f_1(n) = \binom{n}{100} = \frac{n!}{(n-100)!\,100!} = \frac{(n-99)(n-98)\ldots n}{100!} \approx O(n^{100})$$

$\Rightarrow f_3(n) < f_4(n) < f_5(n) < f_2(n) \approx f_1(n)$

Group 2:

$$f_1(n) = 2^{2^{1000000}}, f_2(n) = 2^{100000n}, f_3(n) = \binom{n}{2}, f_4(n) = n\sqrt{n}$$

Xét $f_1(n)$ và $f_2(n)$, ta có: $f_1(n) = O(1)$ (hằng số), $f_2(n) = O(2^n)$

$$\rightarrow f_1(n) < f_2(n)^{(1)}$$

Xét $f_3(n)$ và $f_4(n)$, ta có: $f_3(n) = \binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{(n-2)!(n-1)n}{2!(n-2)!} = \frac{n(n-1)}{2!} = O(n^2)$

$$f_4(n) = n\sqrt{n} = n^{\frac{3}{2}} = O\left(n^{\frac{3}{2}}\right)$$

$$\rightarrow f_4(n) < f_3(n)^{(2)}$$

Xét $f_2(n)$ và $f_3(n)$, ta có: $f_2(n) = O(2^n), f_3(n) = O(n^2)$

$$\rightarrow f_3(n) < f_2(n)^{(3)}$$

Xét $f_1(n)$ và $f_4(n)$, ta có: $f_1(n) = O(1), f_4(n) = O\left(n^{\frac{3}{2}}\right)$

$$\rightarrow f_1(n) < f_4(n)^{(4)}$$

Xét $f_2(n)$ và $f_4(n)$, ta có: $f_2(n) = O(2^n), f_4(n) = O\left(n^{\frac{3}{2}}\right)$

$$\rightarrow f_4(n) < f_2(n)^{(5)}$$

Từ $^{(1)}$, $^{(2)}$, $^{(3)}$, $^{(4)}$ và $^{(5)}$ => $f_1(n) < f_4(n) < f_3(n) < f_2(n)$

Group 3:

$f_1(n) = n^{\sqrt{n}} = 2^{O\left(n^{\frac{1}{2}+c}\right)}$ $(c \approx 10^{-5})$
$f_2(n) = 2^n = 2^{O(n)}$
$f_3(n) = n^{10}.2^{\frac{n}{2}} = 2^{10logn+\frac{n}{2}} = 2^{O(n)}$
$f_4(n) = \sum_{i=1}^{n}(i+1) = \frac{(n+2)(n-1)}{2} = \frac{n^2+n-2}{2} = 2^{\log(n^2+n-2)-1} = 2^{O(n^c)}$

$$\rightarrow f_4(n) < f_1(n) < f_2(n) \approx f_3(n)$$

Group 4: $(n-2)!$, $5\lg(n+100)^{10}$, 2^{2n}, $0.001n^4 + 3n^3 + 1$, $\ln^2 n$, $\sqrt[3]{n}$, 3^n

$f_1(n) = ln^2(n) \in (log_e n)^2$
$f_2(n) = 5\,log(n+100)^{10} = 50log(n+100) \in O(\log n)$
ta có $O(log_e n)^2 < O(\sqrt[3]{n})$ => $f_1(n) < f_2(n)$ $(e > 2)$ (1)
$f_3(n) = \sqrt[3]{n} \in O(\sqrt[3]{n})$
ta có $O(\log n) < O(\sqrt[3]{n})$ => $f_2(n) < f_3(n)$ (2) (2)
$f_4(n) = 0.001n^4 + 3n^3 + 1 \in O(n^4)$
ta có $O(\sqrt[3]{n}) < O(n^4)$ => $f_3(n) < f_4(n)$ (3)
$f_5(n) = (n-2)! = O(n)!$
ta có $O(n^4) < O(n)!$ => $f_4(n) < f_5(n)$ (4)
từ (1)(2)(3)(4) => $f_1(n) < f_2(n) < f_3(n) < f_4(n) < f_5(n)$
⇔ $\boldsymbol{ln^2(n) < 5\,log(n+100)^{10} < \sqrt[3]{n} < 0.001n^4 + 3n^3 + 1 < 3^n < 2^{2n} < (n-2)!}$

Bài tập 5: Chứng minh

5a)

- $O(C) = O(1)$ *với* C *là hằng số*

  $\forall\, C \in N$, *chọn* $c = C$, theo định nghĩa Big-O ta có:

$$O(C) = c * 1$$

$$\rightarrow O(C) = O(1)\ \forall\, C \in N$$

5b)

- O(Cf(n)) = O(f(n)) với C là hằng số

Xét hàm bất kỳ $T_1(n) \in f(n)\ \forall n \geq 1$
=> $T_1(n) <= cf(n) =< c * |C| * f(n)\ \forall n \geq 1$
chọn c =1 và $n_0 = 1$
Theo định nghĩa Big-O => $T_1(n) \in O(Cf(n))$ (1)
Xét hàm bất kỳ $T_2(n) \in Cf(n)\ \forall n \geq 1$

$=> T_2(n) \ <= \ c_1 Cf(n) \ \ \forall n \ \geq 1$

chọn $c = c_1 C$ và $n_0 = 1$

Theo định nghĩa Big-O => $T_1(n) \ \in O(f(n))$ (2)

từ (1) , (2) => O(Cf(n)) = O(f(n))

5c)

- Nếu f(n) ∈ O(g(n)) và g(n) ∈ O(h(n)) thì f(n) ∈ O(h(n))

ta có $f(n) \ \in \ O(g(n)) \ => \ f(n) \ <= \ g(n) \ \forall \, n \ \geq 1$ (1)

và $g(n) \in \ O(h(n)) \ => \ g(n) \ <= \ h(n) \ \forall \, n \ \geq 1$ (2)

từ (1) và (2) => f(n) <= h(n) $\qquad \forall \, n \ \geq 1$

chọn c = 1 và $n_0 = 1$

theo định nghĩa big-O => $f(n) \in \ O(h(n))$

5d)

- $\boldsymbol{N\acute{ế}u\ t_1(n) \in O(f(n))}$ **và** $\boldsymbol{t_2(n) \in O(g(n))\ th\grave{ì}\ t_1(n) + t_2(n) \in O(\max\{f(n), g(n)\})}$

Ta có: $t_1(n) \in O(f(n)) \rightarrow t_1(n) \leq f(n) \ \forall \, n \geq 1^{(1)}$

$t_2(n) \in O(g(n)) \rightarrow t_2(n) \leq g(n) \ \forall \, n \geq 1^{(2)}$

Từ $^{(1)}$ và $^{(2)}$ chọn c = 2 và $n_0 = 1$, theo định nghĩa Big-O ta có:

$$t_1(n) + t_2(n) \leq c * \max\{f(n), g(n)\}$$

$$\rightarrow t_1(n) + t_2(n) = O(\max\{f(n), g(n)\})$$

Bài tập 6: Chứng minh

a) If $\boldsymbol{t(n) \in O(g(n))}$, then $\boldsymbol{g(n) \in \Omega(t(n))}$

$t(n) \in O(g(n)) \Rightarrow \exists c_1 \in R^+, \exists n_{01} \in N$ sao cho $\forall n \geq n_{01}, t(n) \leq c_1 g(n)$

Suy ra: $\forall n \geq n_{01}, \ g(n) \geq \frac{1}{c_1} t(n)$

Chọn: $c = \frac{1}{c_1}, n_0 = n_{01}$

Theo định nghĩa Big-Ω, ta được $g(n) \in \Omega(t(n))$ (đpcm)

b) $\boldsymbol{\Theta(\alpha g(n)) = \Theta(g(n))}$, where $\boldsymbol{\alpha > 0}$

Chứng minh: $\boldsymbol{\Theta(\alpha g(n)) \subset \Theta(g(n))}$

Giả sử: $\forall a(n) \in \Theta(\alpha g(n))$

Suy ra: $\exists c_{01}, c_{02} \in R^+, \exists n_{01} \in N$

$\rightarrow \forall n \geq n_{01}, c_{01}.\alpha g(n) \leq a(n) \leq c_{02}.\alpha g(n)$

Hay: $\forall n \geq n_{01}, \ c_{01}\alpha. g(n) \leq a(n) \leq c_{02}\alpha. g(n)$

Chọn: $c_1 = c_{01}\alpha, c_2 = c_{02}\alpha, n_0 = n_{01}$

Theo định nghĩa Big-Θ, ta được $a(n) \in \Theta(g(n))$

Vậy: $\forall a(n) \in \Theta(\alpha g(n)), a(n) \in \Theta(\mathrm{g}(n))$

$\Rightarrow \Theta(\alpha g(n)) \subset \Theta(g(n))$ (*)

Chứng minh: $\boldsymbol{\Theta(g(n)) \subset \Theta(\alpha g(n))}$

Giả sử: $\forall b(n) \in \Theta(g(n))$

Suy ra: $\exists c_{01}, c_{02} \in R^+, \exists n_{02} \in N$

$\rightarrow \forall n \geq n_{02}, c_{01}. g(n) \leq b(n) \leq c_{02}. g(n)$

Hay: $\forall n \geq n_{02}, \ c_{01}. \alpha g(n) \leq b(n) \leq c_{02}. \alpha g(n)$

Chọn: $c_1 = c_{01}, c_2 = c_{02}, n_0 = n_{02}$

Theo định nghĩa Big-Θ ta được $\mathrm{b}(n) \in \Theta(\alpha g(n))$

Vậy: $\forall b(n) \in \Theta(g(n)), b(n) \in \Theta(\mathrm{\alpha g}(n))$

$\Rightarrow \Theta(g(n)) \subset \Theta(\alpha g(n))$ (**)

Từ (*) và (**), suy ra: $\Theta(\alpha g(n)) = \Theta(g(n))$ (đpcm)

c) $\boldsymbol{\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))}$

Chứng minh: $\boldsymbol{\Theta(g(n)) \subset O(g(n)) \cap \Omega(g(n))}$

Giả sử: $\forall f(n) \in \Theta(g(n))$

Suy ra: $\exists c_1, c_2 \in R^+, \exists n_0 \in N \rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n) \quad \forall n \geq n_0$

Thấy: $\forall n \geq n_0, c_1 g(n) \leq f(n) \Rightarrow f(n) \in \Omega(g(n))$

$\forall n \geq n_0, f(n) \leq c_2 g(n) \Rightarrow f(n) \in O(g(n))$

$\Rightarrow f(n) \in O(g(n)) \cap \Omega(g(n))$

Vậy: $\forall f(n) \in \Theta(g(n)),\ f(n) \in O(g(n)) \cap \Omega(g(n))$

$\Rightarrow \Theta(g(n)) \subset O(g(n)) \cap \Omega(g(n))$ (*)

Chứng minh: $\mathbf{O}(g(n)) \cap \boldsymbol{\Omega}(g(n)) \subset \boldsymbol{\Theta}(\mathbf{g(n)})$

Giả sử: $\forall f(n) \in O(g(n)) \cap \Omega(g(n))$

Suy ra: $\exists c_1 \in R^+, \exists n_{01} \in N \rightarrow f(n) \leq c_1 g(n) \quad \forall n \geq n_{01}$

$\exists c_2 \in R^+, \exists n_{02} \in N \rightarrow c_2 g(n) \leq f(n) \quad \forall n \geq n_{02}$

$\Rightarrow \forall n \geq \max\{n_{01}, n_{02}\}, c_2 g(n) \leq f(n) \leq c_1 g(n)$

$\Rightarrow f(n) \in \Theta(\mathrm{g(n)})$

Vậy: $\forall f(n) \in O(g(n)) \cap \Omega(g(n)),\ f(n) \in \Theta(\mathrm{g(n)})$

$\Rightarrow O(g(n)) \cap \Omega(g(n)) \subset \Theta(\mathrm{g(n)})$ (**)

Từ (*) và (**), suy ra: $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$ (đpcm)

6d)

$\mathbf{max}\{\boldsymbol{f(n), g(n)}\} = \boldsymbol{\theta}(\boldsymbol{f(n) + g(n)})$

Ta có: $t_1(n) \in \theta(f(n)) \rightarrow \frac{1}{2} f(n) \leq t_1(n) \leq f(n)\ \forall\ n \geq 1^{(1)}$

$t_2(n) \in \theta(g(n)) \rightarrow \frac{1}{2} g(n) \leq t_2(n) \leq g(n)\ \forall\ n \geq 1^{(2)}$

Từ $^{(1)}$ và $^{(2)}$ chọn c = 1 và $n_0 = 1$, theo định nghĩa Big-O ta có:

$$\max\{f(n), g(n)\} \leq c * (t_1(n) + t_2(n))$$

$$\rightarrow \max\{f(n), g(n)\} = O(t_1(n) + t_2(n))^{(3)}$$

Từ $^{(1)}$ và $^{(2)}$ chọn c = $\frac{1}{2}$ và $n_0 = 1$, theo định nghĩa Big-Ω ta có:

$$c * (t_1(n) + t_2(n)) \leq \max\{f(n), g(n)\}$$

$$\rightarrow \max\{f(n), g(n)\} = \Omega(t_1(n) + t_2(n))^{(4)}$$

Từ $^{(3)}$ và $^{(4)}$ => $\max\{f(n), g(n)\} = \theta(f(n) + g(n))$

$\log_3(n^2) = \Theta \log_2(n^3)$

6e)

Ta có $2log_3 n$ <= $c\,3log_2 n\ \forall\ n\ \geq 1$

chọn c = 1 và $n_0 = 1$

theo định nghĩa big-O => $log_3 n^2 = O(log_2 n^3)$ (1)

Ta có $2log_3 n$ >= $c\,3log_2 n\ \forall\ n\ \geq 1$

chọn c = $\frac{1}{3}$ và $n_0 = 1$

theo định nghĩa big-Ω => $log_3 n^2 = \Omega(log_2 n^3)$ (2)

theo định nghĩa big- **Θ** từ (1) và (2) => $log_3 n^2 = \Theta(log_2 n^3)$

Bài tập 7: Các khẳng định bên dưới đúng hay sai? Tại sao?

7a)
- Nếu f(n) = Θ(g(n)) và g(n) = Θ(h(n)), thì h(n) = Θ(f(n))

- $f(n) = \Theta(g(n)) \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n) \ \ (\forall n \geq n_0)$
- $g(n) = \Theta(h(n)) \Rightarrow c_3 h(n) \leq g(n) \leq c_4 h(n) \ \ (\forall n \geq n_0)$

$\Longrightarrow c_1 c_3 h(n) \leq f(n) \leq c_2 c_4 h(n)$

$\Longrightarrow \frac{1}{c_2 c_4} f(n) \leq h(n) \leq \frac{1}{c_1 c_3} f(n)$

$\Longrightarrow k_1 f(n) \leq h(n) \leq k_2 f(n)$

Với $\forall n \geq n_0;\ k_1, k_2 > 0 \Longrightarrow h(n) = \Theta(f(n))$ (Khẳng định trên là đúng)

7b)
- Nếu f(n)= O(g(n)) và g(n)= O(h(n)), thì h(n) = Ω(f(n))

- $f(n) = O(g(n)) \Rightarrow f(n) \leq c_1 g(n) \ \ (\forall n \geq n_0)$ (1)
- $g(n) = O(h(n)) \Rightarrow g(n) \leq c_2 h(n) \ \ (\forall n \geq n_0)$ (2)

Từ (1) và (2) $\Longrightarrow f(n) \leq c_1.c_2.h(n) \Longrightarrow h(n) \geq \frac{1}{c_1.c_2} f(n)$

Đặt $k = \frac{1}{c_1.c_2} \Longrightarrow h(n) \geq k.f(n)$

Với $\forall n \geq n_0;\ c_1, c_2 > 0; k > 0 \Longrightarrow h(n) = \Omega(f(n))$ (Khẳng định trên là đúng)

7c)

▪ Nếu f(n)= O(g(n)) và g(n)= O(f(n)), thì f(n)= g(n)

- $f(n) = O(g(n)) \Rightarrow f(n) \leq c_1 g(n)\ \ (\forall n \geq n_0,\ c_1 > 0\ ) \Rightarrow f(n) \subset g(n)$
- $g(n) = O(f(n)) \Rightarrow g(n) \leq c_2 f(n)\ \ (\forall n \geq n_0,\ c_2 > 0\ ) \Rightarrow g(n) \subset f(n)$

$\Longrightarrow f(n) = \mathrm{g(n)} \Longrightarrow$ Khẳng định trên là đúng

7d) $\frac{n}{100} = \Omega(n)$

ta có $\frac{n}{100} >= \frac{1}{100}$*n $\forall\ n\ \geq 1$

chọn c =$\frac{1}{100}$ và $n_0 = 1$

theo định nghĩa big-**Ω** = > $\frac{n}{100} = \Omega(n) \Longrightarrow$ *Khẳng định trên là đúng*

7e)

>$f(n) + O(f(n)) = \theta(f(n))$

Theo định nghĩa Big-O, ta có:

$f(n) \leq O(f(n))\ \forall\ n \geq 1$

$\rightarrow f(n) \leq c * f(n)\ \forall\ n \geq 1, c \in N$

$\rightarrow f(n) + O(f(n)) \geq f(n)\ \forall\ n \geq 1$(1)

Từ (1), ta biết: $f(n) \leq f(n), f(n) \leq O(f(n))\ \forall\ n \geq 1$:

$\rightarrow f(n) + O(f(n)) = O(f(n)) \forall\ n \geq 1$

$\rightarrow f(n) + O(f(n))$ thỏa điều kiện trở thành Big-O của $f(n)$(2)

Do $f(n) + O(f(n))$ luôn không bé hơn $f(n)$

$\rightarrow f(n) + O(f(n))$ không thỏa điều kiện trở thành Big-Ω của $f(n)$(3)

Từ (2) và (3), ta khẳng định được:

$f(n) + O(f(n)) \neq \theta(f(n))$ (dpcm là sai)

7f) $2^{10n} = O(2^n)$

Giả sử: $2^{10n} = O(2^n)$

$\Rightarrow \exists c \in R^+, \exists n_0 \in N \rightarrow 2^{10n} \leq c2^n \quad \forall n \geq n_0$

Lấy logarit cơ số 2 cho hai vế ta được:

$10n \leq logc + n$

$\Rightarrow n \leq \frac{logc}{2} \Rightarrow$ không tồn tại n_0 thỏa $n \geq n_0$

Vậy khẳng định trên là sai

7g) $\log_{10} n = \Theta(\log_2 n)$

Ta có: $\log_{10} n = \log_{10} 2 \log_2 n$

Thấy: $0{,}1.\log_2 n \leq \log_{10} 2\,.\log_2 n \leq 0{,}4 \log_2 n\ \ \forall n \geq 1\quad (\log_{10} 2 \approx 0.301)$

Chọn: $c_1 = 0.1$, $c_2 = 0.4$, $n_0 = 1$

Theo định nghĩa Big-Θ, ta được $\log_{10} n = \Theta(\log_2 n)$

Vậy khẳng định trên là đúng