

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (Data Structures & Algorithms)

L/O/G/O

GV: HUỖNH THỊ THANH THƯỜNG

Email: thuonghtt@uit.edu.vn

HW#03.a: Big-O notation

❖ Bài tập 1:

a). Hãy cho biết ý nghĩa của “độ phức tạp” khi đề cập đến thuật toán?

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 $T(n)$ ”.

c) Nói về ĐPT tức là đề cập tới các ký hiệu tiệm cận, mà có nhiều ký hiệu khác nhau. Vậy khi nào(trong trường hợp nào) thì nên dùng ký hiệu nào?

PS: viết ngắn gọn nhưng đủ ý, viết dài dòng lê thê cô đọc mệt

1-1 Comparison of running times

For 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.

Bài tập 2

Lưu ý: cần giải thích cách làm (cách tính toán) chứ không phải chỉ điền số vào bảng. Không cần giải thích hết các hàm, chỉ lấy 1 hàm nào đó làm ví dụ cũng được

Notes for Chapter 1

15

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
\sqrt{n}							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

HW#03.a: Big-O notation

❖ Bài tập 3:

a) Phép suy ra bên dưới là đúng hay sai và vì sao?

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

HW#03.a: Big-O notation

❖ Bài tập 3:

b) Xét $f(n) = 7n^2$,
 $g(n) = n^2 - 80n$,
 $h(n) = n^3$

Chứng minh:

$$\begin{aligned}f(n) &= O(g(n)), \\g(n) &= O(f(n)), \\f(n) &= O(h(n)), \\h(n) &\neq O(f(n))\end{aligned}$$

HW#03.a: Big-O notation

❖ Bài tập 3:

c) Chứng minh

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

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

$$n \notin O(\log_2 n)$$

HW#03.a: Big-O notation

❖ **Bài tập 4:** Với mỗi nhóm hàm bên dưới, hãy sắp xếp tăng dần theo bậc tăng trưởng Big-O

Group 1:

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

Group 2:

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

Ký hiệu: \log là log cơ số 2, $\binom{n}{k}$ là tổ hợp chập k của n

HW#03.a: Big-O notation

Group 3:

$$\begin{aligned}f_1(n) &= n^{\sqrt{n}} \\f_2(n) &= 2^n \\f_3(n) &= n^{10} \cdot 2^{n/2} \\f_4(n) &= \sum_{i=1}^n (i + 1)\end{aligned}$$

Group 4:

$$\begin{aligned}f_1(n) &= n^4 \binom{n}{2} \\f_2(n) &= \sqrt{n} (\log n)^4 \\f_3(n) &= n^{5 \log n} \\f_4(n) &= 4 \log n + \log \log n \\f_5(n) &= \sum_{i=1}^n i\end{aligned}$$

Group 5:

$$\begin{aligned}f_6(n) &= n^{\sqrt{n}} \\f_7(n) &= n^{\log n} \\f_8(n) &= 2^{n/2} \\f_9(n) &= 3^{\sqrt{n}} \\f_{10}(n) &= 4^{n^{1/4}}\end{aligned}$$

HW#03.a: Big-O notation

Bài tập 5: Chứng minh

- $O(C) = O(1)$ với C là hằng số
- $O(Cf(n)) = O(f(n))$ với C là hằng số
- Nếu $f(n) \in O(g(n))$ và $g(n) \in O(h(n))$ thì $f(n) \in O(h(n))$
- Nếu $t_1(n) \in O(f(n))$ và $t_2(n) \in O(g(n))$ thì $t_1(n) + t_2(n) \in O(\max\{f(n), g(n)\})$

HW#03.b: Các ký hiệu tiệm cận khác

❖ Bài tập 6: Chứng minh

If $t(n) \in O(g(n))$, then $g(n) \in \Omega(t(n))$.

$\Theta(\alpha g(n)) = \Theta(g(n))$, where $\alpha > 0$.

$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$.



$\max\{f(n), g(n)\} = \Theta(f(n) + g(n))$

$f(n) + g(n) = \Theta(\max(f(n), g(n)))$

HW#03.b: Các ký hiệu tiệm cận khác

❖ Bài tập 7: Các khẳng định bên dưới là đúng hay sai? Vì sao?

- Nếu $f(n) = \Theta(g(n))$ và $g(n) = \Theta(h(n))$, thì $h(n) = \Theta(f(n))$
- Nếu $f(n) = O(g(n))$ và $g(n) = O(h(n))$, thì $h(n) = \Omega(f(n))$
- Nếu $f(n) = O(g(n))$ và $g(n) = O(f(n))$, thì $f(n) = g(n)$
- $n/100 = \Omega(n)$
- $f(n) + O(f(n)) = \Theta(f(n))$
- $2^{10n} = O(2^n)$
- $\log_{10} n = \Theta(\log_2 n)$

HW#03.a: Big-O notation

❖ Bài tập 8 (bonus-không bắt buộc): chứng minh các tính chất sau:

- $n + n^2 O(\ln n) = O(n^2 \ln n)$
- $g(n) \in O(h(n)) \Rightarrow O(g(n)) \subseteq O(h(n))$
- $O(f(n)) = O(g(n)) \Leftrightarrow g(n) \in O(f(n)) \text{ và } f(n) \in O(g(n))$
- $O(f(n)) \subset O(g(n)) \Leftrightarrow f(n) \in O(g(n)) \text{ và } g(n) \notin O(f(n))$
- $f(n) \in O(n) \Rightarrow 2^{f(n)} \in O(2^n)$

$f(n) = n^3 + O(n^2)$
means
 $f(n) = n^3 + h(n)$
for some $h(n) \in O(n^2)$

$n^2 + O(n) = O(n^2)$
means
for any $f(n) \in O(n)$:
 $n^2 + f(n) = h(n)$
for some $h(n) \in O(n^2)$