COMP2119: Ch1: Analysis of algorithms

- measure time and space used without implementation
  - not care about difficulty of implementation, only care if it can be implemented.

- measure time cost: count # of primitive operations : $+ - \times \div$, $> < \geq \leq$ (statements)
- only care about huge data
- look at growth rate of program. (ignore constant) (coeff.)
- only count dominating part ( $100n^2 + n \rightarrow n^2$ )

asymptotic running time

Theoretical analysis $\updownarrow$ practical (care const.)

Best / worst / average case: usually consider worst: "upper bound"

average case: $\sum prob(J) \times f_A(J)$

$f_A(J)$: function of $\boxed{n}$

---

Big Theta $\Theta$ : same groth rate "asymptotic tight bound"

$n^2 + 100n = \Theta(n^2)$

$\exists\, c_1, c_2, n_0 > 0$ s.t. $c_1 g(n) \leq f(n) \leq c_2 g(n)$ $\forall\, n \geq n_0$

proof: ( belongs to)

Q: show that $100n^2 + 20n + 5 = \Theta(n^2)$

3 let $c_1 = 1$, $c_2 = 1000$, $n_0 = 1$

2 $c_1 \leq 100 + \frac{20}{n} + \frac{5}{n^2} \leq c_2$

1 $c_1 n^2 \leq 100n^2 + 20n + 5 \leq c_2 n^2$ for all $n \geq n_0$

4 $\therefore 100n^2 + 20n + 5 = \Theta(n^2)$

( worst case only )

proof (not belongs to)

Show that $n^2 \notin \Theta(n^3)$

Assume there exist $c_1, c_2, n_0 > 0$ s.t.

$$c_1 n^3 \le n^2 \le c_2 n^3 \quad \text{for all } n \ge n_0.$$

$$c_1 \le \frac{n^2}{n^3} \le c_2$$

$c_1 \le \frac{1}{n} \qquad n \le \frac{1}{c_1}$

Take $n > \max\{n_0, \frac{1}{c_1}\}$

then $n > \frac{1}{c_1}$ and $n \le \frac{1}{c_1}$

Contradiction

$c_1 \le \frac{n^3}{n^2} \le c_2$

$c_1 \le n \le c_2$

take $n > \max\{n_0, c_2\}$

then $n > c_2$ and $n \le c_2$

Contradiction.

---

Big O: Asymptotic upper bound: There exist $c_1, n_0 > 0$ s.t. $0 \le f(n) \le c_1 g(n)$ for all $n \ge n_0$

(worst → all cases)

$O(f(n)) + O(g(n)) = O(f(n)+g(n))$
$O(f(n))\,O(g(n)) = O(f(n)\,g(n))$
$c = O(1) \qquad n^m = O(n^p)_{m \le p}$

Big Ω: Asymptotic lower bound: There exist $c_1, n_0 > 0$ s.t. $0 \le c_1 g(n) \le f(n)$ for all $n \ge n_0$

(best → all cases)

---

$6n^3 + 10n^2 + 4 = O(n^2)$ X $\quad = \Omega(n^2)$ ✓ $\quad = \Theta(n^2)$ X

$= O(n^3)$ ✓ $\quad = \Omega(n^3)$ ✓ $\quad = \Theta(n^3)$ ✓

$= O(n^4)$ ✓ $\quad = \Omega(n^4)$ X $\quad = \Theta(n^4)$ X

$f(n) = \Theta(g(n))$ iff $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$