9/13/18

→ Worst-Case:

- big-Oh: $T(n) = O(\overbrace{g(n)}^{n^2})$ upper bound $\boxed{"\leq"}$

  $n \in O(n^2)$

- big-Omega: $T(n) \in \Omega(\overbrace{g(n)}^{n^2})$ Lower bound $\boxed{">"}$

  $n^2 \in \Omega(n)$

- Big-Theta: $T(n) \in \Theta(g(n))$ Asymptotic Equation $\boxed{"="}$

Def: $T(n) \in \Theta(g(n))$

$T(n) \in O(g(n))$ and $T(n) \in \Omega(g(n))$

Analogy: $a \geq b$ and $a \leq b \Rightarrow a = b$

Def: $T(n) \in \Omega(g(n))$ means

$\exists\, c > 0,\ n_0 > 0$ such that $\forall\, n \geq n_0$ $T(n) \geq c \cdot g($

→ Example: $T(n) = pn^2 + qn + r$ where $p, q, r \geq 0$

Claim: $T(n) \in \Omega(n^2)$

Proof: Let $c = p-1$ $n_0 = 1$

$pn^2 + qn + r \overset{?}{\geq} (p-1)n^2$ for all $n \geq n_0$.

We can use $c = p$ instead of $c = p-1$

$pn^2 + qn + r \in \Theta(n^2)$

↳ polynomial of highest n^{th} degree $= \Theta(n^{th}$ degree$)$

→ <u>Transitive Property</u>:

If $f(n) \in O(g(n))$ and $g(n) = O(h(n))$
then $f(n) \in O(h(n))$
(Same for Ω)

<u>Analogy</u>:
$a \le b$ & $b \le c$
$\Downarrow$
$a \le c$

→ <u>Inverse Property</u>:

$f(n) \in O(g(n)) \Longleftrightarrow g(n) \in \Omega(f(n))$

$a \le b \Leftrightarrow b \geq a$

- Can't always think in terms of numbers:

$\forall a, b$ ~~a6~~ $a \le b$ or $b \le a$

But, $\exists f(n), g(n)$ such that $f(n) \notin O(g(n))$
$f(n) \notin \Omega(g(n))$

$\Downarrow$ example

$\Downarrow$ <u>Note</u>: Can use = instead of $\in$

?

$$f(n) = \begin{cases} n^2 & n \text{ is odd} \\ 100 & n \text{ is even} \end{cases}$$

$f(n) \in O(n)$? ⟶ $f(n) = n^2$ when odd so can't find no. So no, $f(n) \notin O(n)$.

$f(n) \in \Omega(n)$? ⟶ $f(n)$ is constant when n is even so can't find a no. So no, $f(n) \notin \Omega(n)$.

→ <u>Sum</u>: If $f(n) \in O(h_1(n))$ and $g(n) \in O(h_2(n))$ then
$f(n) + g(n) \in O(h_1(n) + h_2(n))$

(Same for Ω)

<u>Analogous to</u>: $a \le b$ and $c \le d \Rightarrow a + c \le b + d$

→ <u>Products</u>: Apply similarly.

$a \le b$ & $c \le d$
$a \cdot c \le b \cdot d$

Running times of algorithms

→ **Limit Theorem:** $f(n), g(n) > 0$

includes 0

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \begin{cases} < \infty & \text{then } f(n) \in O(g(n)) & \text{CASE 1} \\ > 0 & \text{then } f(n) \in \Omega(g(n)) & \text{CASE 2} \\ 0 < c < \infty & \text{then } f(n) \in \Theta(g(n)) & \text{CASE 3} \end{cases}$$

includes ∞

doesn't include 0 and ∞

Other way: Limit must exist.

→ **Claim:** $\log(n^2) \in \Theta(\log(n)+5)$

$\lim_{n\to\infty} \frac{\log(n^2)}{\log(n)+5}$ Use L'Hopital's Rule!

⇓

If f, g are differentiable such that $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$ or

then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$

$\therefore \lim_{n\to\infty} \frac{\log(n^2)}{\log(n)+5} = \frac{2n \times 1/n^2}{1/n} = \frac{2/n}{1/n} = 2$

Case 3 of limit theorem.

→ Polynomial $\in O$(any exponential)

Any exponential $\notin O$(any polynomial)

? What does it mean to have $f(n) \in O(1)$?

- Upper bounded by constant time.
- In GS algorithm, every step in while loop is $O(1)$

→ Worst-Case:

→ ◦ Look at Running time of a particular algorithm.

- To show $\Theta(n^2)$ need to show $O(n^2)$ and $\Omega(n^2)$

$T(n)$ = (worst-case) running time of GS

$T(n) \in O(n^2)$ → upper bound on while loop

Why do we show only upper bound and not lower bound?

- Need to set up the rankings such that the while loop takes say $\frac{1}{2}n^2$ iterations then

$T(n) \in \Omega(n^2)$

→ ◦ Running time any algorithm for this problem (difficulty of problem itself)