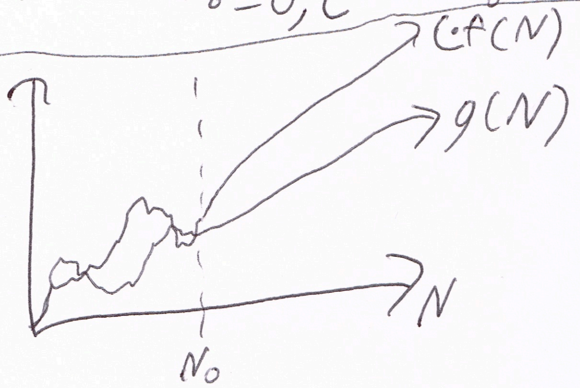
Asymptotic Analysis

$g(N)$ is $O(f(N))$

$\equiv$ g is $O(f)$

if $\exists N_0 \geq 0, c$ $\forall N \geq N_0$ $g(N) \leq c f(N)$



ex. $g(N) = pN^2 + qN + r$

is $g(N)$ $O(N^3)$?   i.e. $f(N) = N^3$

- yes

is $g(N)$ $O(N^2)$?

- yes

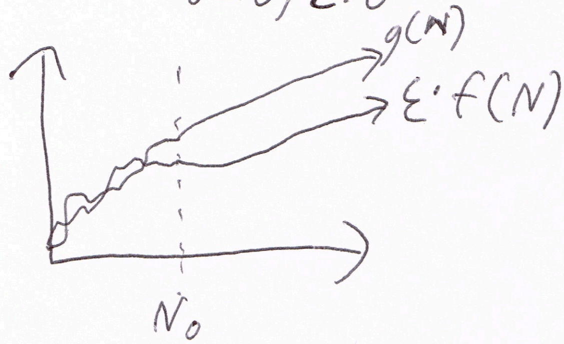
is $g(N)$ $O(N)$?

- no

$g(N)$ is $\Omega(f(N))$ $\qquad$ $g = \Omega(f)$

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

if $\exists N_0 \geq 0, \varepsilon > 0 \ \forall N \geq N_0 \ g(N) \geq \varepsilon \cdot f(N)$



ex. $g(N) = pN^2 + qN + r$

is $g(N)$ $\Omega(N^3)$?

- no

is $g(N)$ $\Omega(N^2)$?

- yes

is $g(N)$ $\Omega(N)$?

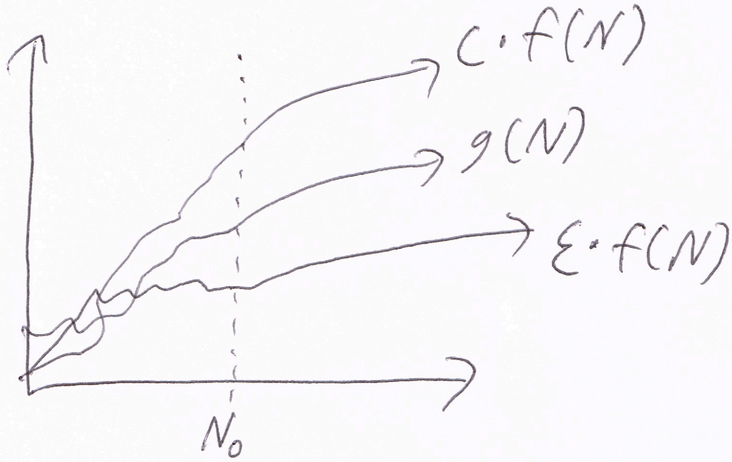
- yes

$g(N)$ is $\Theta(f(N))$

if $g(N)$ is both $O(f(N))$

and

$\Omega(f(N))$



ex. $g(N) = pN^2 + qN + r$

is $g(N)$ $\Theta(N^3)$?

- no

is $g(N)$ $\Theta(N^2)$?

- yes

is $g(N)$ $\Theta(N)$?

- no

properties apply to all O, Ω, Θ

transitivity: if $((g \text{ is } O(h)) \wedge (h \text{ is } O(f)))$

then g is $O(f)$

Sum of functions: g is $O(h)$
f is $O(h)$
then $g+f$ is $O(h)$

multiplication of functions: g is $O(h_1)$
f is $O(h_2)$
then $g \cdot f$ is $O(h_1 \cdot h_2)$

revisit O, Ω

$f(N)$ is $O(g(N))$

$$\equiv \lim_{N \to \infty} \frac{f(N)}{g(N)} = c \vee 0 \neq \infty$$

$f(N)$ is $\Omega(g(N))$

$$\equiv \lim_{N \to \infty} \frac{f(N)}{g(N)} = c \vee \infty \neq 0$$

$f(N)$ is $\theta(g(N))$

$$\equiv \lim_{N \to \infty} \frac{f(N)}{g(N)} = c$$

$g(N) = pN^2 + qN + r$

$$\lim_{N \to \infty} \frac{pN^2 + qN + r}{N^3 + N^2 + 3N} = 0 \qquad g(N) \text{ is } O(N^3)$$

$$\lim_{N \to \infty} \frac{pN^2 + qN + r}{N^2} = p \qquad g(N) \text{ is } \theta(N^2)$$

$$\lim_{N \to \infty} \frac{pN^2 + qN + r}{N} = \infty \qquad g(N) \text{ is } \Omega(N)$$

$T(N) = \max$ # of steps taken by the algorithm on all inputs I of size n.

Case 1: $T(n) \leq U(n)$ then $T(n)$ is $O(U(n))$

show: $\forall$ inputs I of size n,

# of steps taken is $\leq U(n)$

i.e. $T(n) \leq U(n)$ for all I.

Case 2: $T(n) \geq L(n)$ then $T(n)$ is $\Omega(L(n))$

$\exists$ an input I of size n

s.t. # steps $\geq L(N)$

i.e. $T(n) \geq L(N)$ for some I

---

givin $(a_1, a_2, \ldots, a_n)$ find v output i if $a_i = v$

-1 if no such i exists

```
≤n =Θ(n) {for(i=1:n){
Θ(1)        { if(a_i==v)
≤2          {     output i;
≥1
=c          }
O(1) {return -1;
```

$\Theta(n)$

$\Theta(n) \cdot \Theta(1) + \Theta(1) = \Theta(n)$