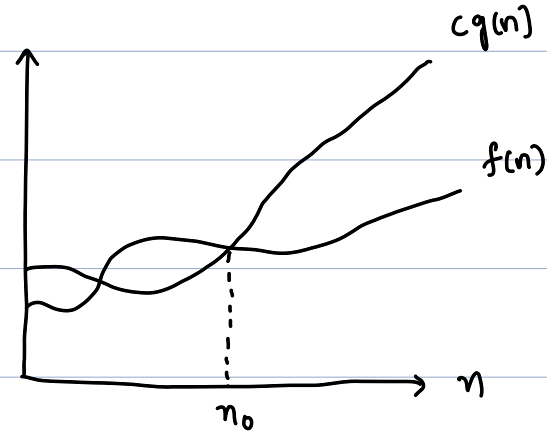
O- notation (read as big oh)

Informally $O(g(n))$ is the set of all functions with a lower or same order of growth as $g(n)$.

Formally

$$O(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le c \cdot g(n) \text{ for all } n \geq n_0 \}$$

ie, For all values $n \geq n_0$, the value of the function $f(n)$ is at most $cg(n)$.



$f(n) \in O(g(n))$.

Notation: If $f(n)$ is a member of the set $O(g(n))$ then we use $f(n) = O(g(n))$ instead of $f(n) \in O(g(n))$.

Example ① $100n + 5 = O(n)$

$$100n + 5 \leq 100n + 5n = 105n$$

Take $\boxed{c = 105,\ n_0 = 1,}$ ie, $100n + 5 \leq 105n \quad \forall\, n \geq 1$

Example: ② $2n^2 = O(n^3)$

$$\boxed{2n^2 \leq 2n^3.}$$

Take constants $\boxed{c = 2,\ n_0 = 1}$

ie, $2n^2 \leq 2n^3 \quad \forall\, n \geq 1$

Example ③ $2n^2 + 3n + 1 = O(n^2)$

$$2n^2 + 3n + 1 \leq 2n^2 + 3n^2 + n^2 = 6n^2$$

take $\boxed{c = 6,\ n_0 = 1}$

ie, $2n^2 + 3n + 1 \leq 6n^2 \quad \forall\, n \geq 1.$

Example 4: $\boxed{4n^2+5 \neq O(n)}$

Suppose there exist constants c and n_0 such that

$$4n^2+5 \leq c.n \quad \forall\ n \geq n_0$$

$$n^2 \leq 4n^2 \leq 4n^2+5 \leq cn$$

ie, $n^2 \leq cn$

$$c \geq n \quad \forall\ n \geq n_0$$

When $n \geq c+1$, $c \not\geq c+1$

which is a contradiction.

Example 5: $f(n) = a_k n^k + \cdots + a_1 n + a_0$ then

$f(n) = O(n^k)$. $\quad a_k \neq 0$

Sol. Choose $n_0 = 1$ and $\boxed{c = |a_k| + \cdots + |a_0|}$.

Need to show $\forall\ \boxed{n \geq 1}$

$$f(n) \leq c.n^k$$

$$\begin{aligned} f(n) &= a_k n^k + \cdots + a_1 n + a_0 \\ &\leq |a_k| n^k + \cdots + |a_1| n + |a_0| \\ &= |a_k| n^k + \cdots + |a_1| n^k + |a_0| n^k \\ &= (|a_k| + \cdots + |a_0|) n^k = c.n^k \quad \forall\ n \geq 1. \end{aligned}$$

//

Example 6 For every $k \geq 1$, n^k is not $O(n^{k-1})$

Proof: by Contradiction

Suppose $n^k = O(n^{k-1})$

By defn $\exists$ $c, n_0 > 0$ such that

$$n^k \leq c \cdot n^{k-1} \quad \forall n \geq n_0$$

$$\underbrace{n \leq c}_{\text{False}} \quad \forall n \geq n_0$$

False. [all integers are not bounded by constant]

You can think of it is like a game.

You want to Prove that inequality $f(n) \leq c\, g(n)$ holds

Your opponent must show that it doesn't hold.

You Pick c, n_0 and opponent Pick any $n \geq n_0$.

you win : $f(n) \leq c\, g(n) \quad \forall n \geq n_0$

else your opponent win.

Game: Player A Selects c, n_0

Player B selects $n \geq n_0$ then

Player A wins if $f(n) \leq c\, g(n) \ \forall\, n \geq n_0$

else Player B wins.

Eg: $f(n) = n^2$, $g(n) = n$, Is $f(n) = O(g(n))$?

Player A selects c, n_0

Player B Pick any n, with $n \geq c+1$

Big-O Visualization

$O(g(n))$ is the set of functions with smaller or same order of growth as $g(n)$

$O(n)$

$3n+5$

$10n$

$1.5n-99$

3

$O(n^2)$

$4n^2+99$

$0.7n^2+2n+1$

$n+2$

45

$O(n\log n)$

$2n\log n+5n$

$10n\log n-98$

$n-3$

101

$O(1)$

4

10

20

Exercise Arrange the following functions in ascending order of growth rate.

If $g(n)$ immediately follows fun $f(n)$ in the ordering, then it should be the case that $f(n) = O(g(n))$

$f_1(n) = n^2$, $f_2(n) = n+2$, $f_3(n) = n^{1/3}$,

$f_4(n) = n \log_2 n$, $f_5(n) = \sqrt{n}$, $f_6 = 2^n$

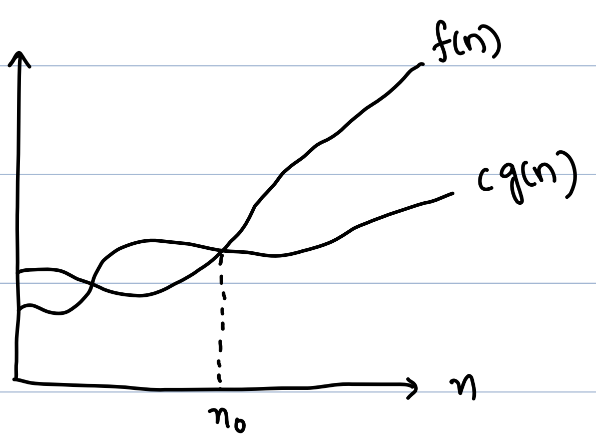
Ω-notation (Big-Omega)

Informally $\Omega(g(n))$ is the set of all functions with larger or same order of growth as $g(n)$.

Formally

$$\Omega(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le c\,g(n) \le f(n) \text{ for all } n \geq n_0 \}$$

ie, For all values $n \geq n_0$, the value of the function $f(n)$ is at least $c\,g(n)$.



$f(n) = \Omega(g(n))$.

Example 1: $5n^2 = \Omega(n)$

we want
$$cn \le 5n^2 \quad \forall \; n \geq n_0$$

take $c = 1$ and $n_0 = 1$ then

$$n \le 5n^2 \quad \forall \; n \geq 1 .$$

Example 2: $n^2+2 = \Omega(n+40)$

We want $\quad c(n+40) \le n^2+2 \quad$ for all $n \geq n_0$

take $c = 1$, $n_0 = 10$ then

$$c(n+40) \le n^2+2 \quad \text{for all } n \geq 10$$

or

take $c = \frac{1}{10}$, $n_0 = 2$ then

$$\frac{1}{10}(n+40) = \frac{n}{10} + 4 \le n^2+2 \quad \forall \; n \geq 2$$

So the constants c and n_0 may not be unique.

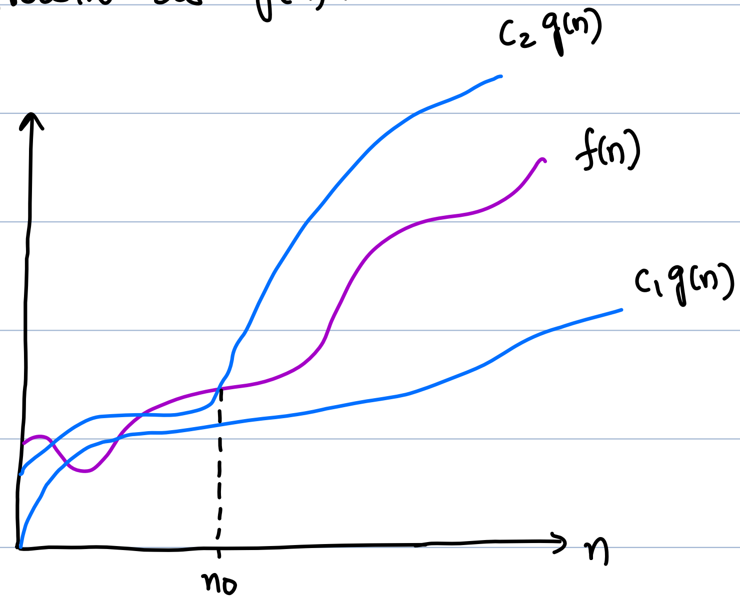
Θ-notation: (Theta)

$\Theta(g(n)) = \{ f(n) :$ there exist positive constants

c_1, c_2 and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$

for all $n \geq n_0 \}$

$\Theta(g(n))$ is the set of functions with the same order of growth as $g(n)$.



$f(n) = \Theta(g(n))$

Example 1 $\quad n+10 = \Theta(n)$

We want

$$c_1 n \le n+10 \le c_2 n \quad \forall\ n \geq n_0$$

take $c_1 = 1$, $c_2 = 11$, $n_0 = 1$ then

$$n \le n+10 \le 11n \quad \forall\ n \geq 1.$$

Example 2: $\quad an^2+bn+c = \Theta(n^2)$; a, b, c are constants & $a>0$.

We want $\quad c_1 n^2 \le an^2+bn+c \le c_2 n^2 \quad \forall\ n \geq n_0$

take $c_2 = |a|+|b|+|c|$

then $an^2+bn+c \le (|a|+|b|+|c|)n^2$

$c_1 = |a|-|b|-|c|$

then $c_1 n^2 \le an^2+bn+c$. $\quad$ and $n_0 = 1$

$$\therefore\ c_1 n^2 \le an^2+bn+c \le c_2 n^2 \quad \forall\ n \geq 1.$$

o- Notation (Little -o)

$o(g(n)) = \{ f(n) :$ for any Positive Constant $c > 0$, there exists a Constant $n_0 > 0$ such that

$$0 \le f(n) < c\, g(n) \quad \forall \; n \geq n_0 \}$$

Example ① $2n = o(n^2)$

We want $2n < c\, n^2$

$$2 < c\, n$$

$$\frac{2}{c} < n$$

Given a $c > 0$ Choose $n_0 > \frac{2}{c}$ then

$2n < c \cdot n^2$ for all $n \geq n_0$

Example ② $2n^2 \neq o(n^2)$

Take $c = \frac{1}{2}$ $\quad 2n^2 < \frac{1}{2} \cdot n^2$

Then there is NO $n_0 > 0$ such that $2n^2 < \frac{1}{2} n^2$ for all $n > n_0$.

Some text books use the following limit definition of the o-notation.

$f(n) = o(g(n))$ means

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Take the example $2n = o(n^2)$ then

$$\lim_{n \to \infty} \frac{2n}{n^2} = 0$$

ω-notation:

$\omega(g(n)) = \{ f(n) :$ for any Positive constant $c>0$,

there exists a constant $n_0 > 0$ such that

$0 \leq c\,g(n) < f(n)$ for all $n \geq n_0 \}$

Example 1: $n^2 = \omega(n)$

we want $cn < n^2$ for any c,

$c < n$

Choose $n_0 > c$ so that $cn < n^2$ $\forall\, n \geq n_0$.

Example 2: $n \neq \omega(n^2)$

$cn^2 < n$ for any c

$cn < 1$

$c < \frac{1}{n}$

Given $c < \frac{1}{n}$ then there is NO n_0 such that

$cn^2 < n$ $\forall\, n \geq n_0$

Hence $n \neq \omega(n^2)$

$f(n) = \omega(g(n))$ means

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

Take the example $2n^2 = \omega(n)$ then

$$\lim_{n\to\infty} \frac{2n^2}{n} = \infty .$$

- Anology between asymptotic comparison of functions f & g and the real numbers a and b

$f(n) = O(g(n))$ is like $a \leq b$

$f(n) = \Omega(g(n))$ is like $a \geq b$

$f(n) = \Theta(g(n))$ is like $a = b$

$f(n) = o(g(n))$ is like $a < b$

$f(n) = \omega(g(n))$ is like $a > b$

Comparing functions

Let $f(n)$ and $g(n)$ are asymptotically Positive

Reflexive : $\boxed{O, \Omega, \Theta}$ are reflexive

ie, $f(n) = O(f(n))$, - - - . .

Transitivity: $\boxed{O, \Omega, \Theta, o, \omega}$ are transitive .

Symmetry: Θ is symmetric.

Notation

Logarithms

$\lg n = \log_2 n$

$\ln n = \log_e n$

$\lg^k n = (\lg n)^k$

$\lg \lg n = \lg(\lg n)$

For $a>0$, $b>0$, $c>0$ and n

(1) $a = b^{\log_b a}$

(2) $a^{\log_b c} = c^{\log_b a}$

① ⓐ $n! = o(n^n)$

ⓑ $n! = \omega(2^n)$

ⓒ $\lg(n!) = \Theta(n\log n)$

ⓐ $$\lim_{n\to\infty} \frac{n!}{n^n} = 0$$

ⓑ $$\lim_{n\to\infty} \frac{n!}{2^n} = \infty$$

ⓒ $$\lg n! = \Theta(n\lg n) = \Theta(\lg n^n)$$

$$0 \le c_1 \lg n^n \le \underbrace{\lg n! \le c_2 \lg n^n}$$

take $c_2 = 1$

$\lg n! \le \lg n^n \quad \forall\, n \geq 1$

Lowerbound :

$$\lg(n!) = \lg 1 + \lg 2 + \dots \lg\left(\tfrac{n}{2}\right) + \dots + \lg n$$

$$\geq \lg\left(\tfrac{n}{2}\right) + \lg\left(\tfrac{n}{2}+1\right) + \dots + \lg(n-1) + \lg n$$

$$\geq \lg\left(\tfrac{n}{2}\right) + \lg \tfrac{n}{2} + \dots + \lg n/2 + \lg n/2 \geq \tfrac{n}{2} \log\left(\tfrac{n}{2}\right)$$

ie, $\lg n! \geq \frac{n}{2}\log n - \frac{n}{2}$

We want to show this is greater than a multiple of $n\log n$

Pick $c_1 = \frac{1}{4}$

$$\frac{1}{4} n\log n < \frac{n}{2}\log n - \frac{n}{2}$$

$$\log n \geq 2$$

$$\frac{1}{4}\log n \geq \frac{1}{2}$$

$$\frac{1}{4} n\log n \geq \frac{n}{2}$$

$$\frac{1}{4} n\log n - \frac{1}{2}n \geq 0$$

$$\frac{1}{2} n\log n - \frac{1}{2}n \geq \frac{1}{4} n\log n$$

Iterated logarithm function :-

$\lg^* n$ (read as "log star of n") to denote the iterated logarithm, defined as

$$\lg^* n = \min\{\, i \geq 0 : \lg^{(i)} n \leq 1 \,\}$$

↳ logarithm applied i times in succession.

① $\lg^* 2 = 1$

② $\lg^* 4 = 2$ $\longrightarrow$ $\lg^{(1)} 4 = 2$

$\lg^{(2)} 4 = \lg(\lg 4) = \lg(2) = 1$

③ $\lg^* 16 = 3$

$\lg^{(1)} 16 = 4 > 1$

$\lg^{(2)} 16 = \lg(\lg(16)) = \lg(4) = 2 > 1$

$\lg^{(3)} 16 = \lg \lg \lg 16 = \lg \lg(4) = \lg(2) = 1$

④ $\lg^* 65536 = 4$

⑤ $\lg^* 2^{65536} = 5$

Exercise: ① Rank the following functions by order of growth.

$n^2,\ n!,\ (1.5)n,\ n^3,\ 2^n,\ e^n,\ n\lg n$

Exercise ② Rank the following functions by order of growth.

$n^{2.5},\ \sqrt{2n},\ n+10,\ 10^n,\ 100^n,\ n^2\lg n$

Exercise ③ $2^{\sqrt{\log n}},\ 2^n,\ n^{4/3},\ n(\lg n)^3,\ n^{\lg n},\ 2^{2^n},\ 2^{n^2}.$

Exercise ④: k is a constant.

$$\binom{n}{k} = \Theta(n^k)$$

Exercise ⑤ Solve Problem 3-3 and 3-4 in Coremen's text book (Pages 61, 62)

Note: Big-O expresses an upperbound not the exact growth rate of the function.

For example, running time of insertion sort is $O(n^2)$, it is also correct to say that running time of insertion sort is $O(n^3)$.

But we prefer to make them tight as possible.

Example:- Searching array A (of length n) for an integer t ?

```
Search(A, n, t)
for i=1 to n
    if A[i] == t
        return TRUE
return FALSE
```

Running time = $O(n)$

Example: Search in arrays A and B (of lengths n.) for an integer t.

```
Search(A, B, n, t)
for i=1 to n
    if A[i] == t
        return TRUE
for i=1 to n
    if B[i] == t
        return TRUE
return FALSE
```

Running time: $O(n)$

Example : Given an arrays A, B, Does they have a number in common?

```
Common(A, B, n)
For i = 1 to n
    For j = 1 to n
        if A[i] = = B[j]
            return TRUE
return FALSE.
```

Running time: $O(n^2)$

Example Given an arrays A, Does A have any repeated elements

```
For i = 1 to n
    For j = i+1 to n
        if A[i] = = A[j]
            return TRUE
return FALSE.
```

Running time: $O(n^2)$