

Asymptotic - Analysis

O- notation (read as big oh)

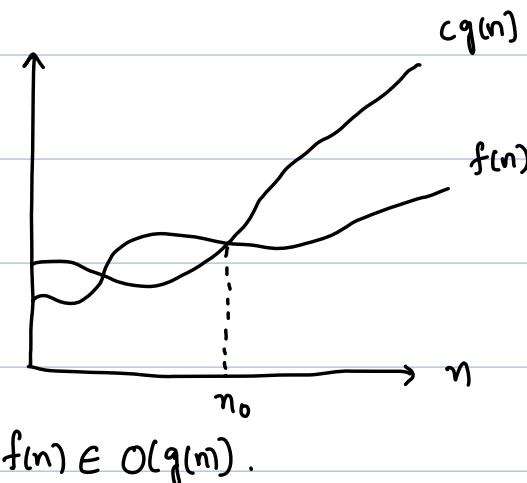
Informally $O(g(n))$ is the set of all functions

with a lower or same order of growth as $g(n)$.

Formally

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

i.e., for all values $n \geq n_0$, the value of the function $f(n)$ is at most $c g(n)$.



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 $C = 105$, $n_0 = 1$, ie,

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

$\forall n \geq 1$

Example ②

$$2n^2 = O(n^3)$$

$$2n^2 \leq 2n^3.$$

Take constants $C = 2$, $n_0 = 1$

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

Example ③

$$2n^2 + 3n + 1 = O(n^2)$$

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

take $C = 6$, $n_0 = 1$

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

Example 4:

$$4n^2 + 5 \neq O(n)$$

Suppose there exist constants c and n_0 such that

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

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

$$\text{ie, } n^2 \leq cn$$

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

$$\text{When } n \geq c+1, \quad c \not\geq c+1$$

which is a contradiction.

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

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

Sol: Choose $n_0 = 1$ and $c = |a_k| + \dots + |a_0|$.

Need to show $\forall n \geq 1$

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

$$f(n) = a_k n^k + \dots + a_1 n + a_0$$

$$\leq |a_k| n^k + \dots + |a_1| n + |a_0|$$

$$= |a_k| n^k + |a_1| n^k + |a_0| n^k$$

$$= (|a_k| + \dots + |a_0|) n^k = c \cdot n^k \quad \forall n \geq 1. \quad //$$

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

Pf: by contradiction

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

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

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

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



False. [all integers are not bounded by constant]

You can think of it is a game.^{like}

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 $n > n_0$.

If you win: $f(n) \leq c g(n) \quad \forall n > n_0$

Else your opponent win.

Game: Player A selects c, n_0

Player B selects $n \geq n_0$ then

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

else Player B wins.

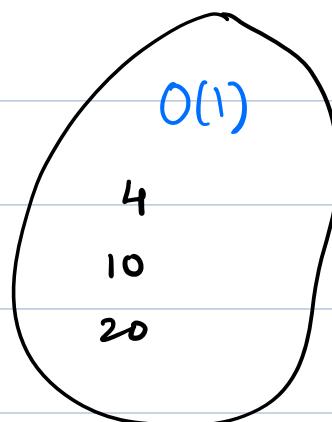
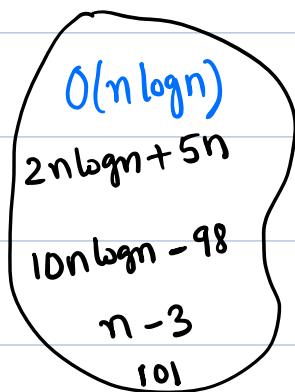
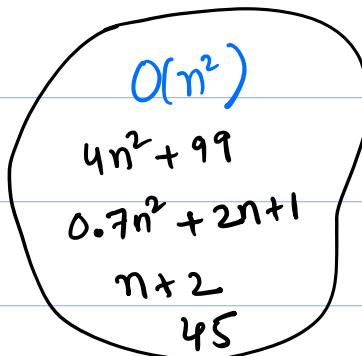
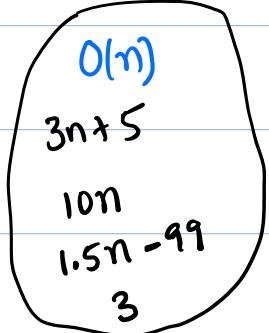
Eg: $f(n) = n^2, g(n) = n$, Is $f(n) = O(g(n))$?

Player A selects c, n_0

Player B Pick any n , $n \geq c+1$ ^{With}

Big-O Visualization

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



Exercise

Arrange the following functions in ascending order of growth rate.

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

$$f(n) = O(g(n))$$

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

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

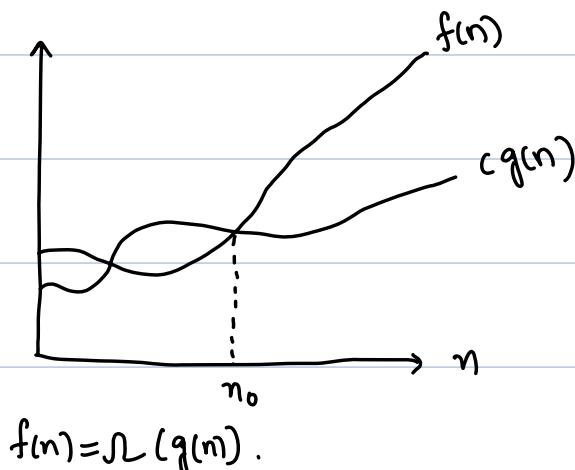
Ω -notation (Big-Omega)

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

Formally

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

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



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

We want

$$cn \leq 5n^2 \text{ if } n \geq n_0$$

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

$$n \leq 5n^2 \text{ if } n \geq 1.$$

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

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

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

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

(or)

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

$$\frac{1}{10}(n+40) = \frac{n}{10} + 4 \leq n^2+2 \text{ if } n \geq 2$$

So the constants c and n_0 may not be unique.

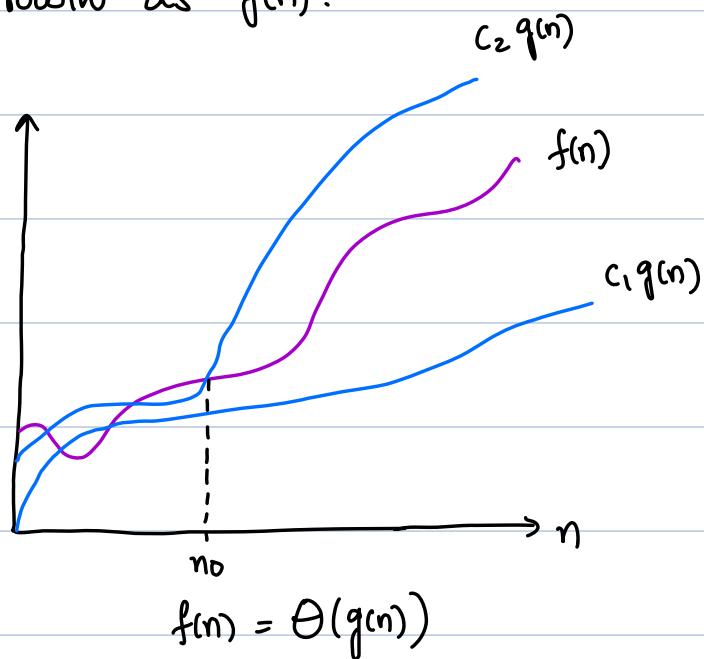
Θ -notation: (Theta)

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

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

for all $n \geq n_0\}$

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



Example 1

$$n+10 = \Theta(n)$$

We want

$$c_1 n \leq n+10 \leq c_2 n \quad \forall n > n_0$$

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

$$n \leq n+10 \leq 11n \quad \forall n > 1.$$

Example 2:

$$an^2 + bn + c = \Theta(n^2) ; a, b, c \text{ are constants} \& a > 0.$$

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

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

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

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

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

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

O-Notation (Little-O)

$O(g(n)) = \{ f(n) : \text{for any Positive Constant } C > 0,$

there exists a constant $n_0 > 0$ such that

$$0 \leq f(n) < c g(n) \quad \forall n > n_0 \}$$

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

$$2n < c \cdot n^2 \quad \text{for all } n > n_0$$

Example ②

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

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

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 \mathcal{O} -notation.

$f(n) = \mathcal{O}(g(n))$ means

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

Take the example $2n = \mathcal{O}(n^2)$ then

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

ω -notation:

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

there exists a constant $n_0 > 0$ such that

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

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

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

$$c < n$$

choose $n_0 > c$ so that $cn < n^2$ if $n \geq n_0$.

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

$$cn^2 < n \text{ for any } c$$

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

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

$$cn^2 < n \text{ if } n \geq n_0$$

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

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

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

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

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

- Analogy 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: O, Ω, Θ are reflexive

i.e., $f(n) = O(f(n))$, ...

Transitivity: $O, \Omega, \Theta, o, \omega$ 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

$$\textcircled{1} \quad a = b^{\log_b a}$$

$$\textcircled{2} \quad a^{\log_b c} = c^{\log_b a}$$

$$\textcircled{1} \quad @ \quad n! = O(n^n)$$

$$\textcircled{2} \quad n! = \omega(2^n)$$

$$\textcircled{3} \quad \lg(n!) = \Theta(n \lg n)$$

$$\textcircled{4} \quad \lim_{n \rightarrow \infty} \frac{n!}{n^n} = 0$$

$$\textcircled{5} \quad \lim_{n \rightarrow \infty} \frac{n!}{2^n} = \infty$$

$$\textcircled{6} \quad \lg n! = \Theta(n \lg n) = \Theta(\lg n^n)$$

$$0 \leq c_1 \lg n^n \leq \lg n! \leq c_2 \lg n^n$$

take $c_2 = 1$

$$\lg n! \leq \lg n^n + n > 1$$

Lowerbound :

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

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

$$\geq \lg\left(\frac{n}{2}\right) + \lg\frac{n}{2} + \dots + \lg n_k + \lg n_k \geq \frac{n}{2} \lg\left(\frac{n}{2}\right)$$

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

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

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

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

$$\log n \geq 2$$

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

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

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

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

Iterated logarithm function :-

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

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

\hookrightarrow logarithm applied i times
in succession.

$$\textcircled{1} \quad \lg^* 2 = 1$$

$$\textcircled{2} \quad \lg^* 4 = 2$$

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

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

$$\textcircled{3} \quad \lg^* 16 = 3$$

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

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

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

$$\textcircled{4} \quad \lg^* 65536 = 4$$

$$\textcircled{5} \quad \lg^* 2^{65536} = 5$$

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

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)

TRUE | FALSE

① If $f(n) = O(g(n))$ then $g(n) = O(f(n))$

② If $f(n) = O(g(n))$ then $g(n) = \Omega(f(n))$

Multiple choice

① Consider the functions n^k , c^n where $k \geq 1$, $c > 1$

are constants, then

a) $n^k = O(c^n)$

b) $n^k = \Omega(c^n)$

c) $n^k = \Theta(c^n)$

② Which of the following is/are True

a) $n^3 \log_2 n = O(3n \log_2 n)$

b) $n^3 \log_2^n = \Omega(3n \log_2^n)$

c) $n^3 \log_2^n = \Theta(3n \log_2^n)$

③ Which of the following is/are True

a) $8^n = O(4^n)$

b) $8^n = \Omega(4^n)$

c) $8^n = \Theta(4^n)$

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 $A[i] == t$

Running time = $O(n)$

return TRUE

return FALSE

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 array 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: $O(n^2)$