Apriori Analysis

1) $i = n$ **True**

while $i > 2$:

$i = i^{1/25}$ ———

print(i)

$i = n$

$n^{1/25}$ — 1 $\qquad \log_2 2 = 1$

$\left(n^{1/25}\right)^{1/25} \longrightarrow n^{1/25^2}$

$n^{1/25^3}$

$\vdots$

k times

$n^{1/25^k} = 2$

$n^{1/25^k} = 2$

$\log_2 n^{1/25^k} = \cancel{\log_2 2}^{\,1}$

$\frac{1}{25^k} \log_2 n = 1$

$\log_{25} 25 = 1$

$\log_2 n = 25^k$

$\log_{25}(\log_2 n) = k \cancel{\log_{25} 25}^{\,1}$

$$\boxed{k = \log_{25}(\log_2 n)}$$

$\Rightarrow O(\log(\log n))$

2) $i = 29$ $\quad n < n$ — false

while $i < n$:

$i = i^{23}$

$(29)^{23^1}$

$(29)^{23^2}$

$\vdots$

$\log_n (29)^{23^k} = \cancel{\log_n n}\ 1$

$(29)^{23^k} = n$

$$\log_n (29)^{23^k} = 1$$

$$23^k \log_n (29) = 1$$

$$\log_n (29) = 1/23^k$$

$$\boxed{k = \log_{23}(\log_{29} n)}$$

3) $i = 1$ $\quad 36 < 40$

$6 < 40$

while $i < n$:

$i = 2 * i$

$i = 3 * i$

$i = 6 * i$

$\log_6 n$

$n = 40$

$i = 6 * 1$

$i = 6 * 6$

$i = 6 * 36$

Complexity classes

$\log(\log n)$

1) constant – $O(1)$ —— $O(\log(\log n))$
2) Logarithmic – $O(\log n)$
3) Linear – $O(n)$
4) Quadratic – $O(n^2)$
5) cubic – $O(n^3)$
6) Polynomial – $O(n^c)$
7) Exponential – $O(c^n)$ – $O(2^n)$

Increasing order time complexity

$O(n^3)$ — Cubic
$O(3^n)$ — Exponential

$n! = O(n^n)$

8) $n! < n^n$
$2^n < n^n$
$n! > 2^n$

$f(n) \le c \cdot g(n)$

$$\boxed{2^n < n! < n^n}$$

$n = 1000$

$n = 16$

$\Downarrow$

16^{16}

9) $\log n < n$
$(\log n)^2 < n$
$(\log n)^3 < n$
⋮
$(\log n)^{1000} < n$

$$\boxed{(\log n)^{\log n} > n}$$ — True

10) $2^n < 3^n$
$2^n < (2 \times 1.5)^n$

11) $\log_2 n > \log_3 n$ (Lower Base

↳ Higher complexity in Logarithmic)

$O(\log n)$

Asymptotic Notations → **Important**

↳ 1) Big O → worst case scenario
(O)

2) Omega → Best case scenario
(Ω)

3) Theta → Average case scenario
(θ)

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

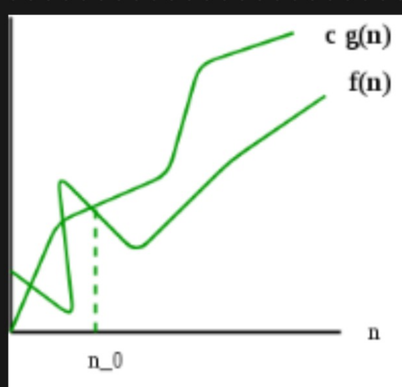
$\Downarrow$

if $f(n) \leq c \cdot g(n)$

$\forall n,\ n \geq n_0$

$c > 0$

$n_0 \geq 1$



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

Example 1

$f(n) = 5n$

$g(n) = n$

→ **True**

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

$\Downarrow$

$\boxed{f(n) \leq c \cdot g(n)}$

constant

↑

$\boxed{5n \leq c \cdot n}$

$C = 5, \geq 5$ → constant

$\log n = O(n) \Longrightarrow$

$\log n \leq n$

Example 2

$f(n) = n$

$g(n) = 5n$

$f(n) = O(g(n)) \rightarrow$ True

$f(n) <= c \cdot g(n)$

$\boxed{n <= c \cdot 5n}$

$c = 1/5$

$n <= n \rightarrow$ True

Example 3

$f(n) = n^2 \quad g(n) = n$

Not hold true $\leftarrow f(n) = O(g(n))$ ✗

$f(n) <= c \cdot g(n)$

$c \propto n$

$n^2 <= c \cdot n$

$c = n$ — Not constant

$n^2 <= n^2$

Problem 1

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

$$\hookrightarrow \quad 1000\, n\log n = O\left(\frac{n\log n}{1000}\right)$$

$$\Downarrow$$

True

$$f(n) <= c \cdot g(n)$$

$$\boxed{1000\, n\log n <= c \cdot \frac{n\log n}{1000}}$$

$$c = 1000000$$

Problem 2

$$\underset{f(n)}{2^{n+5}} = O(\underset{g(n)}{2^n}) \text{ — True}$$

$$f(n) <= c \cdot g(n)$$

$$2^{n+5} <= c \cdot 2^n$$

$$\Downarrow$$

$$\boxed{2^n \cdot 2^5 <= c \cdot 2^n}$$

$$c = 2^5 = 32$$

Problem 3

$$\frac{1}{n} = O(n)$$

True

$$1/n <= c \cdot n$$

$$c = 1$$

Problem 4

$$\boxed{2^{2n} = O(2^n)} \longrightarrow \text{False}$$

$$2^{2n} \propto = c \cdot 2^n$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$2^n \cdot 2^n \qquad\qquad c = 2^n$$

$$\Downarrow$$

c is not constant

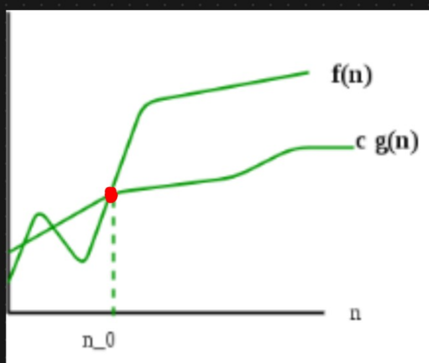
Omega Notation

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

$$f(n) \geq c \cdot g(n) \ \forall n, \ n \geq n_0$$

$$c > 0$$

$$n_0 \geq 1$$



f(n) = Omega(g(n))

Example 1

$$f(n) = n$$

$$g(n) = 5n$$

$$\boxed{f(n) = \Omega(g(n))} \leftarrow \text{True}$$

$$f(n) \geq c \cdot g(n)$$

$$\boxed{n \geq c \cdot 5n}$$

$$c = 1/5$$

Example 2

$$f(n) = 5n, \quad g(n) = n$$

$$f(n) = \Omega(g(n)) \text{ — True}$$

$$f(n) \geq c \cdot g(n)$$

$$\boxed{5n \geq c \cdot n}$$ → True

$c = 5$

Example 3

$f(n) = n$

$g(n) = n^2$

→ False

$$\boxed{f(n) = \Omega(g(n))}$$

$f(n) \geq c \cdot g(n)$

$n \geq c \cdot n^2$

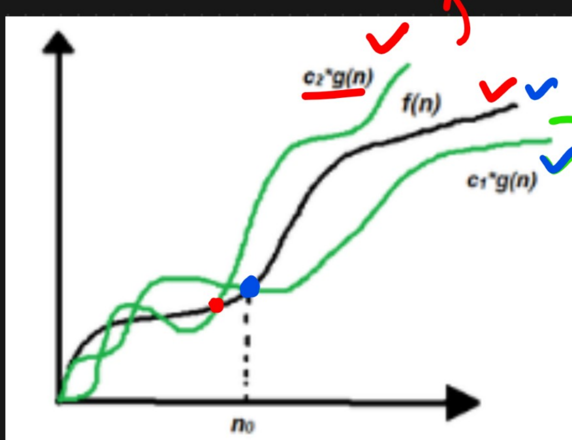
$c = 1/n$

↓

c is not constant

Big O

Theta Notation



↳ Big O $= f(n) <= c_2 g(n)$

AND

Omega $= f(n) >= c_1 g(n)$

Omega

$c_1 \,\&\, c_2 > 0$

Example 1

$f(n) = n$

$g(n) = 5n$

$f(n) = \Theta(g(n))$ ⟶ True

$f(n) <= c_1 \cdot g(n)$ — True

$\boxed{n <= c_1 \cdot 5n}$ (Big O)

$c_1 = 1/5$

$f(n) >= c_2 \cdot g(n)$ — Omega

$\boxed{n >= c_2 \cdot 5n}$

$c_2 = 1/5$

Recurrence Relation → Solve

1) Substitution
2) Recursive Tree
3) Master's Theorem

Practice Problems

1) $f(n) = n - 10$ $\quad$ $g(n) = n + 10$

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

2) $f(n) = n$ $\quad$ $g(n) = n$

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

3) $64^{\log_2 n} \cdot 32^{\log_2 n} = O(n^5)$

4) $\dfrac{4^n}{2^n} = O(2^n)$

5) $128^{\log_2 n} \cdot n^2 = \Theta(n^9)$