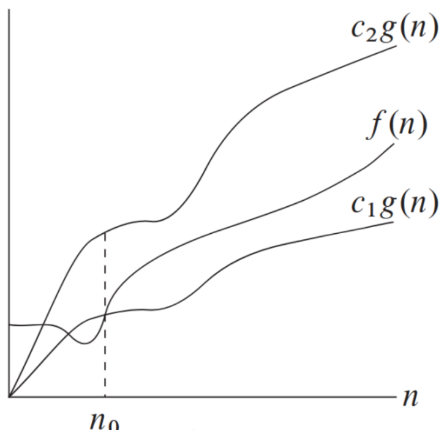
Data Structures and Algorithms
Dr. Javad Vahidi
Session 01 - Time Complexity
October 13, 2023

Exercise 1.
Imagine that you are in a class full of N CS students, you know one of them has the same birthday as you do, define an algorithm to find this person and analyze the "Time Complexity"

Solution 1.
The Answer is to fully search the class and ask one by one so $T(N) \in O(N)$

Θ Notation:

$\Theta(g(n)) = f(n)$: 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 $n \geq n_0$

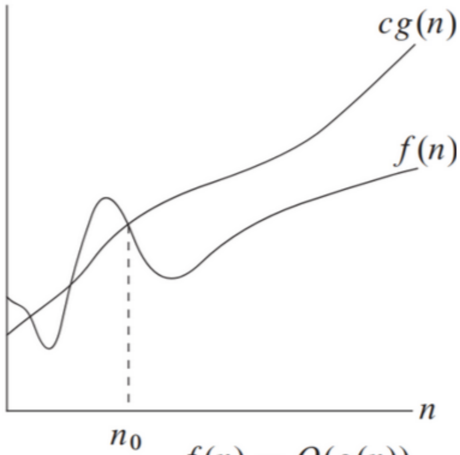


$f(n) = \Theta(g(n))$

note: $f(n)$ could be an algorithm.

Big O Notation:

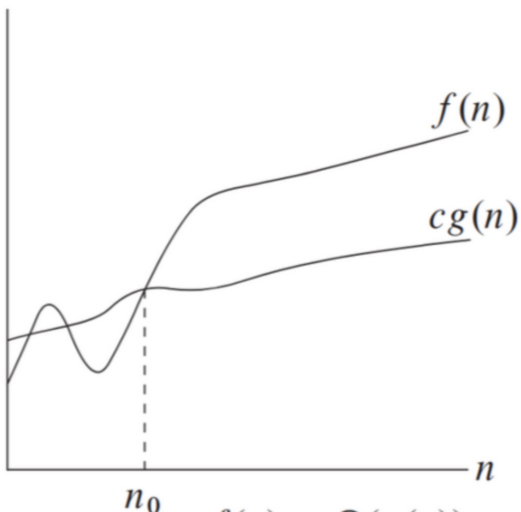
$O(g(n)) = f(n)$: there exist positive constants c and n_0 such that $0 \leq f(n) \leq c * g(n)$ for all $n \geq n_0$



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

Ω Notation:

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



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

Small-o Notation:

$o(g(n)) = f(n)$: for any positive constants c there exist a constatn n_0 such that $0 \leq f(n) \leq c * g(n)$ for all $n \geq n_0$

also:

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

ω Notation:

$\omega(g(n)) = f(n)$: for any positive constants $c > 0$ there exist a constatn $n_0 > 0$ such that $0 \leq c * g(n) < f(n)$ for all $n \geq n_0$

$$f(n) \in \omega(g(n)) iff g(n) \in o(f(n))$$

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

Exercise 2.
Show Transitivity on Big O

Solution 2.
Assume $f(n) \leq c_1 * g(n)$, for all $n \geq k_1$ and $g(n) \leq c_2 * h(n)$ for all $n \geq k_2$, therefore,
$f(n) \geq c_1 * g(n) \geq c_1 * c_2 * h(n)$
$c = max(c_1, c_2), k = max(k_1, k_2)$
$f(n) \leq c * h(n)$ for all $n \geq k_1 \implies f(n) \in O(h(n))$

Exercise 3.
Show that if $f(n) = O(s(n), g(n) = O(r(n)) \implies \frac{f(n)}{g(n)} = O(\frac{s(n)}{r(n)})$

Solution 3.
counter example:

$$f(n) = n^3$$

$$g(n) = n^2$$

Exercise 4.
Show that if $f(n) = O(g(n) \implies g(n) = \Omega(f(n))$

Solution 4.
$O(g(n)) = f(n)$: there exist positive constants c and n_0 such that $0 \leq f(n) \leq c * g(n)$ for all $n \geq n_0$
$g(n) \geq \frac{1}{c} f(n) \implies g(n) = \Omega(f(n))$

Exercise 5.
Show that if $max(f(n), g(n)) = \Theta(f(n) + g(n))$

Solution 5.
To prove this, we have to show that there exists constants $c_1, c_2, n_0 > 0$ such that for all $n \geq n_0$,

$$0 \leq c_1(f(n) + g(n)) \leq max(f(n), g(n)) \leq c_2(f(n) + g(n))$$

$f(n) + g(n) \geq max(f(n), g(n))$
also: $f(n) \leq max(f(n), g(n)) \& g(n) \leq max(f(n), g(n))$
$\implies f(n) + g(n) \leq 2max(f(n), g(n))$
$\implies 0 \leq \frac{1}{2}(f(n) + g(n)) \leq max(f(n), g(n)) \leq f(n) + g(n), n \geq n_0$
$\implies max(f(n), g(n)) = \Theta(f(n) + g(n))$

Exercise 6.

True or False?

a) if $f(n) = \Omega(n^2)$ and $g(n) = \Omega(n) \implies f(g(n)) = \Omega(n^3)$
b) $f(g(n)) = \Theta(g(f(n))$
c) $f(n) = \Theta(f(\frac{n}{2}))$

Solution 6.

a)False! $f(n) = n^2, g(n) = n$
b)False! $f(n) = Ln(n), g(n) = n^2$
c)False! $f(n) = 4^n$