Analysis of algorithms

To analyze the efficiency of an algorithm, we need to measure the "time" needed to execute an algorithm as a function of the size of the input. The "time" or "running time" is really the number of instructions executed. The size of the input may consist of one or more parameters.

Examples of size of input: array of size n □ input size n
set of n objects □ input size n
$m \times n$ matrix □ input size parameters m and n
graph with n vertices and m edges □ input size parameters m and n

The maximum time needed to execute an algorithm is the **worst-case time** for the algorithm.
The minimum time needed to execute an algorithm is the **best-case time** for the algorithm.
The average time needed to execute an algorithm is the **average-case time** for the algorithm.

Running times might be complicated looking polynomials, logarithmic, exponential or factorial functions.

Objective: We would like to be able to order functions by their asymptotic size (from lowest/fastest (1) to highest/slowest (8)).

$10n^2 \lg n + 8n^3$
$80n\sqrt{n}$
$n! + 2^n$
$100n + 5n \lg n$
$15n + 30\sqrt{n}$
$1000n + 10n^2$

Example: Suppose the total number of steps executed in an algorithm with input size n is

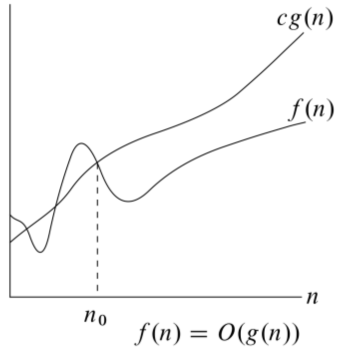
$$t(n) = 30n^2 + 12n + 15$$

n	$t(n) = 30n^2 + 12n + 15$	$30n^2$
10	3,144	3,000
100	301,215	300,000
1000	30,012,015	30,000,000
10000	3,000,120,015	3,000,000,000

As n gets large, $t(n)$ grows like $30n^2$. Furthermore, a change in processor speeds would only affect the running time by a constant factor. (For example, if a new processor was 30 times faster than the old one, then we would take $t(n)$ and divide by 30.) Thus, $t(n)$ grows on the same order of growth as the function n^2. We will next carefully define what this means and write $t(n) = \theta(n^2)$.

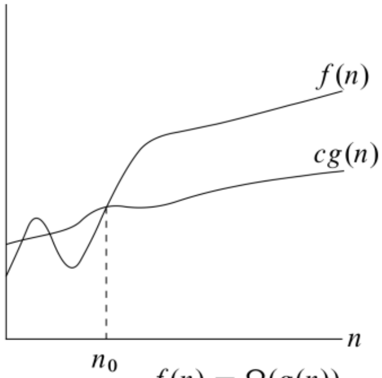
Definition 2.3.2 Let f and g be nonnegative functions on the positive integers.

(a) We write $f(n) = O(g(n))$ and say that $f(n)$ is of order at most $g(n)$ if there exist constants $c > 0$ and n_0 such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$. Equivalently, we say $f(n)$ is big oh of $g(n)$ or g is an asymptotic upper bound for f.



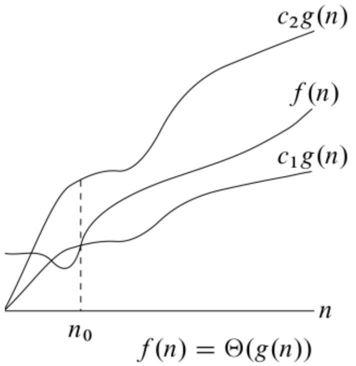
$f(n) = O(g(n))$

(b) We write $f(n) = \Omega\ (g(n))$ and say that $f(n)$ is of order at least $g(n)$ (or $f(n)$ is big omega of $g(n)$ or g is an asymptotic lower bound for f) if there exist constants $c > 0$ and n_0 such that $f(n) \geq c \cdot g(n)$ for all $n \geq n_0$.

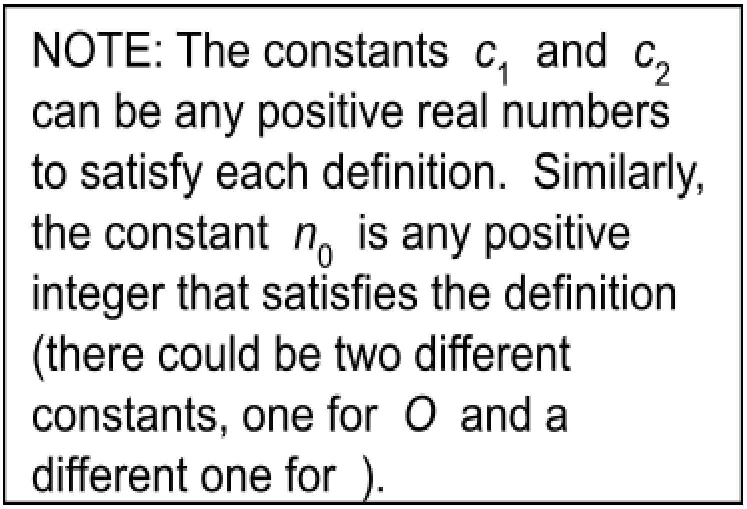


$f(n) = \Omega(g(n))$

(c) We write $f(n) = \theta(g(n))$ and say that $f(n)$ is of order $g(n)$ (or $f(n)$ is big theta of $g(n)$ or g is an asymptotic tight bound for f) if both $f(n) = O(g(n))$ and $f(n) = \Omega\ (g(n))$.



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



NOTE: The constants c_1 and c_2 can be any positive real numbers to satisfy each definition. Similarly, the constant n_0 is any positive integer that satisfies the definition (there could be two different constants, one for O and a different one for).

Applying the Definition

Example 1: Use the definition of O to show that $30n^2 + 12n + 15 = O(n^2)$.
[We need to find a constant c and n_0 such that $30n^2 + 12n + 15 \le c \cdot n^2$ for all $n \ge n_0$.]

Example 2: Use the definition of Ω to show that $30n^2 + 12n + 15 = \Omega(n^2)$.
[We need to find constants

Example 3: Use the definition of Θ to show that $30n^2 + 12n + 15 = \Theta(n^2)$.

Example 4: Use the definition of "big Oh" to show that $2n^3 - 6n = O(n^3)$.

Example 5: Use the definition of "big Omega" to show that $2n^3 - 6n = \Omega(n^3)$.

Example 6: Conclusion?

Example 7: Find the mistakes in the following "proofs".

We will show $2n^3 - 6n = O(n^3)$.
Let N = 1 and C = – 4. For all $n \ge 1$, we know $6n \le 6n^3$. So

$$2n^3 - 6n \le 2n^3 - 6n^3 = -4n^3.$$

Next we will show $2n^3 - 6n = \Omega(n^3)$.
Let N = 1 and C = – 4. For all $n \ge 1$, we know $n \le n^3$. So $-6n \ge -6n^3$.
Thus

$$2n^3 - 6n \ge 2n^3 - 6n^3 = -4n^3.$$

Example 8: Use the definition of O to show that $4n^3 + 2n^2 - 12n + 10 = O(n^3)$.

Example 9: Using the definition of "big Omega", prove that $4n^3 + 2n^2 - 12n + 10 = \Omega\,(n^3)$.

Example 10: Using the definition of "big Omega", prove that $3n^3 + n^2 - 7n + 3 = \Omega\,(n^3)$.