Introduction to Programming and Data Structures

Analysis of Algorithms

Malay Bhattacharyya

Associate Professor

MIU, CAIML, TIH
Indian Statistical Institute, Kolkata

November, 2023

Order of growth

The order of growth of the running time of an algorithm may provide a simple (yet quantitative) characterization of the algorithm's efficiency.

This may allow us to compare the relative performance of alternative algorithms.

The motivation

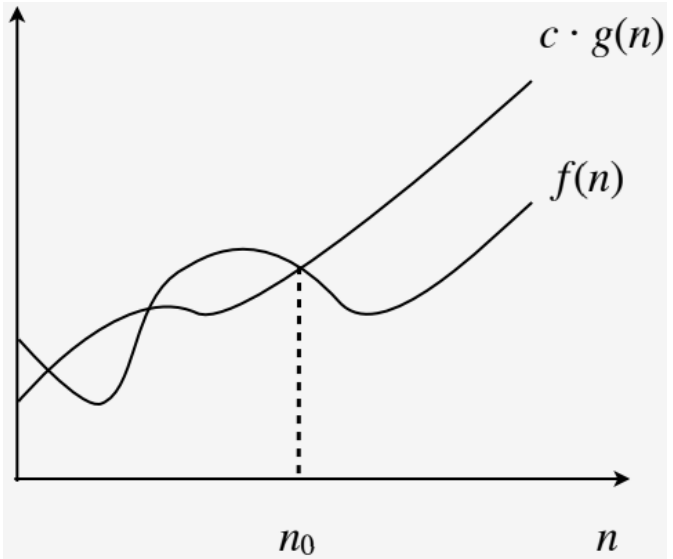
Suppose we want to quantitatively analyze the running time of an algorithm.

- Can we consider the running time as a function of the input size?
- Can we study how the function behaves if we let it run when the input size grows toward the infinity?
- Can we put bounds on these functions?
- How can we say that the function is arbitrarily close to the bound?

Note: The term `asymptotic` refers to approaching a value or curve arbitrarily closely.

O Notation

The function $f(n)$ is $O(g(n))$ if there exist constants $c > 0$ and $n_0 \geq 0$ such that $0 \leq f(n) \leq c * g(n)$ for all $n \geq n_0$.



For example, consider $f(n) = 12n^2 + 5n + 10$.

- $f(n)$ is $O(n^2)$ (non-trivial)
- $f(n)$ is $O(n^2 \log n)$, $O(n^3)$, $O(n^3 \log n)$, $O(n^4)$, ... (trivial)
- $f(n)$ is neither $O(n)$ nor $O(n \log n)$.

Note: $f(n)$ and $g(n)$ are real-valued functions.

O Notation (contd.)

Remember that $O(g(n))$ is a set of functions, but we do often write $f(n) = O(g(n))$ instead of $f(n) \in O(g(n))$. This is a notational abuse and has to be handled carefully.

For example, consider $f_1(n) = 2n^2$ and $f_2(n) = 5n^3$.

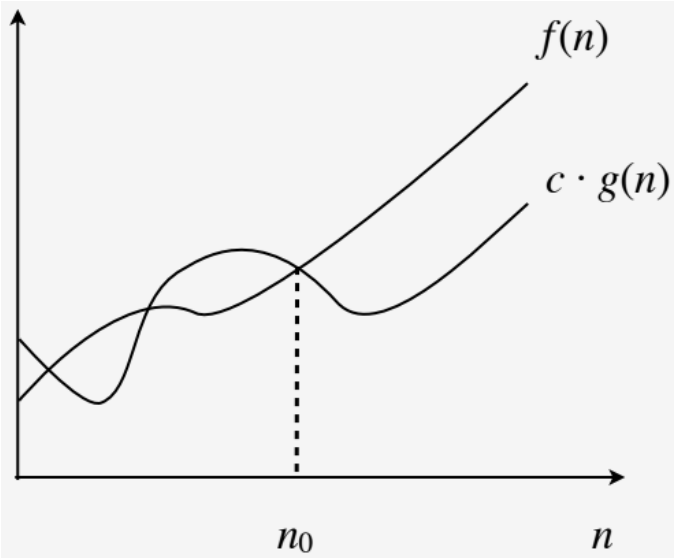
- One can write $f_1(n) = O(n^3)$ and $f_2(n) = O(n^3)$
- However, $f_1(n) = f_2(n)$ can not be concluded

O Notation (contd.)

- $f(n)$ is $O(f(n))$ (reflexivity)
- If $f(n)$ is $O(g(n))$ and $c > 0$, then $c * f(n)$ is $O(g(n))$ (constant bounding)
- If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then $f_1(n) * f_2(n)$ is $O(g_1(n) * g_2(n))$. (multiplicative bounding)
- If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then $f_1(n) + f_2(n)$ is $O(\max g_1(n), g_2(n))$. (additive bounding)
- If $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n)$. (transitivity)

Ω Notation

The function $f(n)$ is $\Omega(g(n))$ if there exist constants $c > 0$ and $n_0 \geq 0$ such that $f(n) \geq c * g(n) \geq 0$ for all $n \geq n_0$.



For example, consider $f(n) = 12n^2 + 5n + 10$.

- $f(n)$ is $\Omega(n^2)$ (non-trivial)
- $f(n)$ is $\Omega(n^0 = 1)$, $\Omega(n)$ (trivial)
- $f(n)$ is not $\Omega(n^3)$.

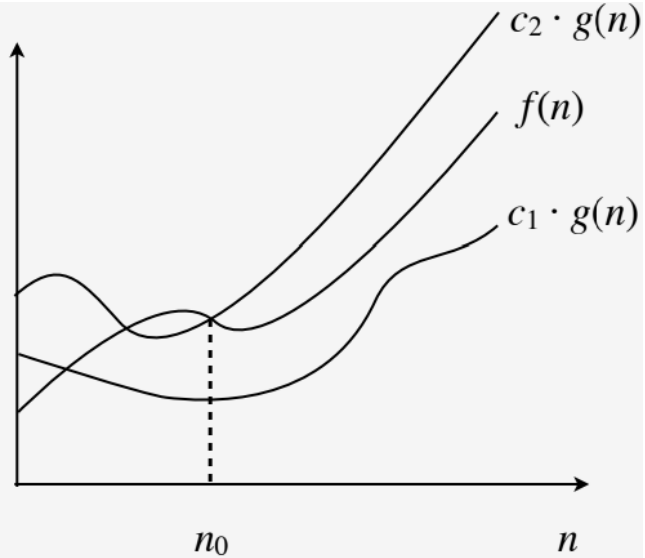
Note: $f(n)$ and $g(n)$ are real-valued functions.

Ω Notation (contd.)

- $f(n)$ is $\Omega(f(n))$ (reflexivity)
- If $f(n)$ is $\Omega(g(n))$ and $c > 0$, then $c * f(n)$ is $\Omega(g(n))$ (constant bounding)
- If $f_1(n)$ is $\Omega(g_1(n))$ and $f_2(n)$ is $\Omega(g_2(n))$, then $f_1(n) * f_2(n)$ is $\Omega(g_1(n) * g_2(n))$. (multiplicative bounding)
- If $f_1(n)$ is $\Omega(g_1(n))$ and $f_2(n)$ is $\Omega(g_2(n))$, then $f_1(n) + f_2(n)$ is $\Omega(\min g_1(n), g_2(n))$. (additive bounding)
- If $f(n)$ is $\Omega(g(n))$ and $g(n)$ is $\Omega(h(n))$, then $f(n)$ is $\Omega(h(n)$. (transitivity)

Θ Notation

The function $f(n)$ is $\Theta(g(n))$ if there exist constants $c_1, c_2 > 0$ and $n_0 \geq 0$ such that
$0 \leq c_1 * g(n) \leq f(n) \leq c_2 * g(n)$ for all $n \geq n_0$.



For example, consider
$f(n) = 12n^2 + 5n + 10$.

- $f(n)$ is $\Theta(n^2)$
- $f(n)$ is neither $\Theta(n \log n)$ nor $O(n^2 \log n)$.

Note: $f(n)$ and $g(n)$ are real-valued functions.

Θ Notation (contd.)

- $f(n)$ is $\Theta(f(n))$ (reflexivity)
- If $f(n)$ is $\Theta(g(n))$ and $c > 0$, then $c * f(n)$ is $\Theta(g(n))$ (constant bounding)
- If $f_1(n)$ is $\Theta(g_1(n))$ and $f_2(n)$ is $\Theta(g_2(n))$, then $f_1(n) * f_2(n)$ is $\Theta(g_1(n) * g_2(n))$. (multiplicative bounding)
- If $f_1(n)$ is $\Omega(g_1(n))$ and $f_2(n)$ is $\Omega(g_2(n))$, then $f_1(n) + f_2(n)$ is $\Omega(\max g_1(n), g_2(n))$. (additive bounding)
- If $f(n)$ is $\Omega(g(n))$ and $g(n)$ is $\Omega(h(n))$, then $f(n)$ is $\Omega(h(n)$. (transitivity)

Problems

1. Suppose $f(n) = n^2 + 139n \log n + 500$. What is the $O()$ of the function?
2. Suppose $f(n) = 10$. What is the $O()$ of the function?
3. Show that $f(n)$ is $\Theta(g(n))$ iff $f(n)$ is both $O(g(n))$ and $\Omega(g(n))$.
4. Let $f(n) = a_0 + a_1 n + \ldots + a_d n_d$ with $a_d > 0$. Then prove that $f(n)$ is $\Theta(n_d)$.