Growth of Functions

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

Big-O

Let $f(n)$ and $g(n)$ be two functions of n.

We say that $f(n)$ **grows asymptotically no faster than** $g(n)$ if there is a constant $c_1 > 0$ such that

$$f(n) \leq c_1 \cdot g(n)$$

holds for all n at least a constant c_2.

We can denote this by $f(n) = O(g(n))$.

Example

Verify all the following:

$$
\begin{aligned}
10000000 &= O(1) \\
100\sqrt{n} + 10n &= O(n) \\
1000n^{1.5} &= O(n^2) \\
(\log_2 n)^3 &= O(\sqrt{n}) \\
(\log_2 n)^{9999999999} &= O(n^{0.0000000001}) \\
n^{0.0000000001} &\neq O((\log_2 n)^{9999999999}) \\
n^{9999999999} &= O(2^n) \\
2^n &\neq O(n^{9999999999})
\end{aligned}
$$

An interesting fact:

$$\log_{b_1} n \quad = \quad O(\log_{b_2} n)$$

for any constants $b_1 > 1$ and $b_2 > 1$.

For example, let us verify $\log_2 n = O(\log_3 n)$.

Notice that

$$\log_3 n = \frac{\log_2 n}{\log_2 3} \Rightarrow \log_2 n = \log_2 3 \cdot \log_3 n.$$

Hence, we can set $c_1 = \log_2 3$ and $c_2 = 1$, which makes

$$\log_2 n \leq c_1 \log_3 n$$

hold for all $n \geq c_2$.

An interesting fact:

$$\log_{b_1} n \quad = \quad O(\log_{b_2} n)$$

for any constants $b_1 > 1$ and $b_2 > 1$.

Because of the above, in computer science, we omit all the constant logarithm bases in big-O. For example, instead of $O(\log_2 n)$, we will simply write $O(\log n)$.

- Essentially, this says that "you are welcome to put any constant base there; and it will be the same asymptotically".

Big-Ω

Let $f(n)$ and $g(n)$ be two functions of n.

If $g(n) = O(f(n))$, then we define:

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

to indicate that $f(n)$ **grows asymptotically no slower than** $g(n)$.

The next slide gives an equivalent definition.

Big-Ω

Let $f(n)$ and $g(n)$ be two functions of n.

We say that $f(n)$ **grows asymptotically no slower than** $g(n)$ if there is a constant $c_1 > 0$ such that

$$f(n) \geq c_1 \cdot g(n)$$

holds for all n at least a constant c_2.

We can denote this by $f(n) = \Omega(g(n))$.

Example

Verify all the following:

$$
\begin{aligned}
\log_2 n &= \Omega(1) \\
0.001n &= \Omega(\sqrt{n}) \\
2n^2 &= \Omega(n^{1.5}) \\
n^{0.0000000001} &= \Omega((\log_2 n)^{9999999999}) \\
\frac{2^n}{1000000} &= \Omega(n^{9999999999})
\end{aligned}
$$

Big-Θ

Let $f(n)$ and $g(n)$ be two functions of n.

If $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, then we define:

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

to indicate that $f(n)$ **grows asymptotically as fast as** $g(n)$.

Example

Verify the following:

$$
\begin{aligned}
10000 + 30\log_2 n + 1.5\sqrt{n} &= \Theta(\sqrt{n}) \\
10000 + 30\log_2 n + 1.5n^{0.5000001} &\neq \Theta(\sqrt{n}) \\
n^2 + 2n + 1 &= \Theta(n^2)
\end{aligned}
$$