Analysis of Algorithms

Alvin Lin

August 2017 - December 2017

Big-Oh Running Time (asymptotic upper bound)

$f(n) = O(g(n))$ if there exists a constant $c > 0$ and a constant $n_\circ$ such that for every $n \geq n_\circ$ we have $f(n) \leq cg(n)$.

Example

Prove that $n^3 = O(7n^2 + \frac{n^3}{3})$: We need to find $c, n_\circ$ such that $\forall n \geq n_\circ$:

$$n^3 \leq c(7n^2 - \frac{n^3}{3})$$

$$c = 3$$

Example

Prove that $n^3 = O(\frac{n^3}{3} - 7n^2)$: We need to find $c, n_\circ$ such that $\forall n \geq n_\circ$:

$$n^3 \geq c(\frac{n^3}{3} - 7n^2)$$

We can take $c = 6$, then we want to show that:

$$\begin{aligned} n^3 &\leq 6(\frac{n^3}{3} - 7n^2) \quad \forall n \geq n_\circ \\ n^3 &\leq 2n^3 - 42n^2 \\ 42n^2 &\leq n^3 \\ 42 &\leq n \end{aligned}$$

Thus, this inequality holds for $n_\circ = 42$.

Example

Prove that $\log_{10} n = O(\log n)$ and that $\log n = O(\log_{10} n)$: We need to find $c, n_\circ$ such that $\forall n \geq n_\circ$:

$$\log_{10} n \leq c \log_2 n$$
$$\log_{10} n = \frac{\log_2 n}{\log_2 10}$$

We can take $c = \frac{1}{\log_2 10}$ and $n_\circ = 1$.

Example

What about 3^n and 2^n?

$$2^n = O(3^n)$$
$$2^n \neq O(2^n)$$

For this case, we need to show that for every possible c and every possible n_0, the equality is false. By contradiction, assume that $3^n = O(2^n)$. If this is the case, then there exists constants c and $n_\circ$ such that 3^n is upper bounded by $c \cdot 2^n$.

$$3^n \leq c \cdot 2^n \quad \forall n \geq n_\circ$$

Divide both sides by 2^n:

$$(\frac{3}{2})^n \leq c$$

This cannot possibly hold as $n \to \infty$ because $\frac{3}{2} > 1$ and the function is exponential.

Big-Omega Running Time (asymptotic lower bound)

$f(n) = \Omega(g(n))$ if there exists a constant $c > 0$ and a constant $n_\circ$ such that for every $n \geq n_\circ$ we have $f(n) \geq cg(n)$. This is the reverse of Big-Oh and determines a lower bound. Claim: any comparison-based sort of n numbers needs at least $\Omega(n \log n)$ comparisons.

Theta Running Time (asymptotically tight bound)

$f(n) = \Theta(g(n))$ if there exists constants $c_1, c_2 > 0$ and a constant $n_\circ$ such that for every $n \geq n_\circ$ we have $c_1 g(n) \leq f(n) \leq c_2 g(n)$.

$$\log n = \Theta(1 + \log n)$$

$$n^3 = \Theta(7n^2 + \frac{n^3}{3})$$

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech