Standard Computer Science Notation and Mathematics

© Quentin F. Stout

$\begin{rcases} \text{lg} & \log \text{ base } 2 \\ \text{ln} & \log \text{ base } e \end{rcases}$ Remember that $\log_a x = \log_a b \cdot \log_b x$.

$n!$ n factorial, i.e., $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$.

$\binom{n}{m}$ n choose m, the number of distinct subsets of m items in a set of n distinct items. It is equal to $n!/[m!(n-m)!]$, or $n(n-1) \cdot \ldots \cdot (n-m+1)/m!$.

$\lfloor x \rfloor$ the floor of x, i.e., the largest integer no greater than x.

$\lceil x \rceil$ the ceiling of x, i.e., the smallest integer no smaller than x.

Some Sums

$$\sum_{i=0}^{n} C\binom{n}{i} x^i y^{n-i} = C(x+y)^n$$

$$\sum_{i=1}^{n} Cx^i = \begin{cases} C(x^{n+1}-x)/(x-1) & x \neq 1 \\ Cn & x = 1 \end{cases}$$

If $n = \infty$ then this is ∞ for $x \geq 1$, undefined for $x \leq -1$, and $Cx/(1-x)$ for $|x| < 1$.

$$\sum_{i=1}^{n} Cix^i = C[nx^{n+2} - (n+1)x^{n+1} + x]/(x-1)^2 \quad x \neq 1$$

To derive this, differentiate the preceding formula and multiply by x. If $n = \infty$ then this is ∞ for $x \geq 1$, undefined for $x \leq -1$, and $Cx/(1-x)^2$ for $|x| < 1$.

$$\sum_{i=1}^{n} Ci = Cn(n+1)/2$$

$$\sum_{i=1}^{n} (a_k i^k + a_{k-1} i^{k-1} + \ldots + a_1 i + a_0) = \frac{a_k}{k+1} n^{k+1} + b_k n^k + \ldots + b_1 n \,,$$

for any positive integer k, and real constants $a_k \ldots a_0$. The constants $b_k \ldots b_1$ depend only on their index, k, and the a's.

Integral Test

One of the most powerful methods available for estimating a variety of sums is known as the *integral test*. You may have used such a test in your calculus classes, though there it is usually used to estimate an infinite sum. Here it will be used to estimate a finite sum.

For a positive function f defined on the non-negative reals,

- if f is nonincreasing, then $\int_0^n f(x)\,dx \; \geq \; \sum_1^n f(i) \; \geq \; \int_1^{n+1} f(x)\,dx$

- if f is nondecreasing, then $\int_0^n f(x)\,dx \; \leq \; \sum_1^n f(i) \; \leq \; \int_1^{n+1} f(x)\,dx$

Therefore, whenever f is either nonincreasing or nondecreasing,

$$\left| \sum_1^n f(i) - \int_1^n f(x)\,dx \right| \leq \int_0^1 f(x)\,dx + \int_n^{n+1} f(x)\,dx$$

O, Ω, ω, o, and Θ Notation
(sometimes called generalized O-notation)

Throughout, f, g, and h are positive functions defined on the nonnegative reals. The following notation is as standard as possible, but many authors use slightly different definitions. In particular, while here it will be written that, say $f \in \Theta(g)$, this more often appears (even in my own writings) as $f = \Theta(g)$. Also, many scientists use O when they could have used Θ. It appears that only computer scientists use Θ.

f is of order g, written $f \in \Theta(g)$ or $f(n) \in \Theta(g(n))$ ("f is big theta of g"), if there are positive constants C, D, and N such that $Cg(n) \leq f(n) \leq Dg(n)$ for all $n \geq N$. I.e., eventually f is trapped between two multiples of g.

f is of order at most g, written $f \in O(g)$ or $f(n) \in O(g(n))$ ("f is big oh of g"), if there are positive constants D and N such that $f(n) \leq Dg(n)$ for all $n \geq N$. I.e., eventually f remains below some multiple of g.

f is of order at least g, written $f \in \Omega(g)$ or $f(n) \in \Omega(g(n))$ ("f is big omega of g"), if there are positive constants C and N such that $Cg(n) \leq f(n)$ for all $n \geq N$. I.e., eventually f remains above some multiple of g.

f is of order less than g, written $f \in o(g)$ or $f(n) \in o(g(n))$ ("f is little oh of g"), if for any positive constant D, there is a positive integer N (depending on D), such that $f(n) \leq Dg(n)$ for all $n \geq N$. I.e., for any multiple of g, no matter how small, f ultimately remains below it.

f is of order greater than g, written $f \in \omega(g)$ or $f(n) \in \omega(g(n))$ ("f is little omega of g"), if for any positive constant C, there is a positive integer N (depending on C), such that $Cg(n) \leq f(n)$ for all $n \geq N$. I.e., for any multiple of g, no matter how large, f ultimately remains above it.

The above can be extended to functions of any number of variables. For example,
$f(n_1, \ldots, n_k) \in \Theta(g(n_1, \ldots, n_k))$ if there are positive constants C, D, and N such that
$Cg(n_1, \ldots, n_k) \leq f(n, \ldots, n_k) \leq Dg(n, \ldots, n_k)$ for all $n_1, \ldots, n_k$ such that $\min_{i=1}^k n_i \geq N$,
i.e., when all of the variables are sufficiently large.

Some Useful Properties

1. $f \in \Theta(g)$ if and only if $f \in O(g)$ and $f \in \Omega(g)$.

2. $f \in O(g)$ if and only if $g \in \Omega(f)$.

3. $f \in o(g)$ if and only if $g \in \omega(f)$.

4. If $f \in o(g)$ then $f \in O(g)$ and $f \notin \Omega(g)$.

5. If $f \in \omega(g)$ then $f \in \Omega(g)$ and $f \notin O(g)$.

6. If $f \in \Theta(h)$, $g \in \Theta(h)$, and there is an $N \geq 0$ such that $f(n) \leq e(n) \leq g(n)$ for all $n \geq N$, then $e \in \Theta(h)$.

7. If $f \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (h)$ and $g \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (h)$ then $a \cdot f + b \cdot g \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (h)$ for all $a, b > 0$.

8. If $f \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (g)$ and $g \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (h)$ then $f \in \left\{ \begin{matrix} \Theta \\ O \\ \Omega \\ o \\ \omega \end{matrix} \right\} (h)$.

9. $f \in o(g)$ if and only if $\lim_{n \to \infty} f(n)/g(n) = 0$.

10. $f \in \omega(g)$ if and only if $\lim_{n \to \infty} f(n)/g(n) = \infty$.

11. If $\lim_{n \to \infty} f(n)/g(n) = a$, $0 < a < \infty$, then $f \in \Theta(g)$. The notation $f \sim g$ is sometimes used to indicate that $\lim_{n \to \infty} f(n)/g(n) = 1$, and hence if $f \sim g$ then $f \in \Theta(g)$.

12. $f \in O(g)$ if and only if $\limsup_{n \to \infty} f(n)/g(n) < \infty$.

13. $f \in \Omega(g)$ if and only if $\liminf_{n \to \infty} f(n)/g(n) > 0$.

14. If $a_k > 0$ then $a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0 \in \Theta(n^k)$.

15. Since $\log_a x = \log_a b \cdot \log_b x$, $\log_a n \in \Theta(\log_b n)$ for all $a, b > 1$, and we typically don't indicate the base of the logarithm inside O notation.

16. $\displaystyle\sum_{i=1}^{n} i^p \in \begin{cases} \Theta(n^{p+1}) & p > -1 \\ \Theta(\log n) & p = -1 \\ \Theta(1) & p < -1 \end{cases}$.   This is easy to prove via the integral test.

17. $\displaystyle\sum_{i=1}^{n} \log_a^p i = \Theta(n \log^p n)$ for $a > 1$ and $p \geq 1$. In particular, $\log_a n! = \sum_{i=1}^{n} \log_a i \in \Theta(n \log n)$. This also follows from Stirling's approximation: $n! \sim \sqrt{2\pi n} \left(\dfrac{n}{e}\right)^n$.

18. If $a > 0$ and $b > 1$ then $\log_b n \in o(n^a)$.

19. If $b > a > 0$ then $n^a \in o(n^b)$.

20. If $c > 1$ and $a > 0$ then $n^a \in o(c^n)$.

21. If $d > c > 0$ then $c^n \in o(d^n)$.

22. If $c > 0$ then $c^n \in o(n!)$.

While the functions occuring in class are nicely behaved, there are pairs of functions which are not comparable using O, Ω, Θ, o, or ω, i.e., there are functions f and g such that $f \notin \Omega(g)$ and $f \notin O(g)$. For example, let $f = |\sin n| + 1/n$, and let $g = |\cos n| + 1/n$.

Some very common, but not universally agreed upon, terminology for rates of function growth are:

> **constant** $\Theta(1)$
>
> **logarithmic** $\Theta(\log n)$
>
> **polylogarithmic** $O(\log^k n)$ for some $k > 0$
>
> **sublinear** $o(n)$
>
> **linear** $\Theta(n)$
>
> **nearly linear** $O(n \log^k n)$ for some $k > 0$
>
> **superlinear** $\omega(n)$
>
> **quadratic** $\Theta(n^2)$
>
> **cubic** $\Theta(n^3)$
>
> **polynomial** or **feasible** $O(n^k)$ for some $k > 0$
>
> **exponential** $\Omega(C^n)$ and $O(D^n)$ for some $1 < C \le D$
>
> **superexponential** $\omega(C^n)$ for any constant C
>
> **doubly exponential** $\Omega\left(C_1^{C_2^n}\right)$ and $O\left(D_1^{D_2^n}\right)$ for some $C_1, C_2, D_1, D_2 > 1$

Unless otherwise stated, these always refer to the time requirements of an algorithm, rather than its space requirements.

Since one often encounters algorithms that are nearly linear, or nearly quadratic, sometimes the notation $\tilde{\Theta}$ is used, where $f = \tilde{\Theta}(g)$ means $f \in \Omega(g(n)/\log^j(n))$ and $f \in O(g(n)\log^k(n))$ for some $j, k \ge 0$. It is called "soft theta".