# Asymptotic Notation Cheat Sheet

Growth of Functions - Quick Reference

## The 5 Notations

| Not. | Meaning | Like |
|---|---|---|
| $\Theta$ | Tight bound | $a = b$ |
| $O$ | Upper bound | $a \leq b$ |
| $\Omega$ | Lower bound | $a \geq b$ |
| $o$ | Strict upper | $a < b$ |
| $\omega$ | Strict lower | $a > b$ |

**$\Theta(g(n))$:** $\exists c_1, c_2, n_0 > 0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ $\forall n \geq n_0$

**$O(g(n))$:** $\exists c, n_0 > 0 : 0 \leq f(n) \leq c \cdot g(n)$ $\forall n \geq n_0$

**$\Omega(g(n))$:** $\exists c, n_0 > 0 : 0 \leq c \cdot g(n) \leq f(n)$ $\forall n \geq n_0$

**$o(g(n))$:** $\forall c > 0, \exists n_0 : 0 \leq f(n) < c \cdot g(n)$ $\forall n \geq n_0$

OR: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

**$\omega(g(n))$:** $\forall c > 0, \exists n_0 : 0 \leq c \cdot g(n) < f(n)$ $\forall n \geq n_0$

OR: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

## Key Theorems

**Th 3.1:** $f(n) = \Theta(g(n)) \iff f(n) = O(g(n))$ AND $f(n) = \Omega(g(n))$

**Polynomial:** $p(n) = \sum_{i=0}^{d} a_i n^i$, $a_d > 0 \implies p(n) = \Theta(n^d)$

**Transpose:** $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$

$f(n) = o(g(n)) \iff g(n) = \omega(f(n))$

## Growth Hierarchy

$$O(1) < O(\lg n) < O(\sqrt{n}) < O(n)$$
$$< O(n \lg n) < O(n^2) < O(n^3)$$
$$< O(2^n) < O(n!)$$

## Common Complexities

| Class | Example |
|---|---|
| $O(1)$ | Array access |
| $O(\lg n)$ | Binary search |
| $O(n)$ | Linear search |
| $O(n \lg n)$ | Merge/Heap sort |
| $O(n^2)$ | Bubble/Insertion sort |
| $O(2^n)$ | Subset generation |
| $O(n!)$ | Permutations |

## Properties

**Transitivity:** $f = \Theta(g), g = \Theta(h) \implies f = \Theta(h)$

(Same for $O, \Omega, o, \omega$)

**Reflexivity:** $f(n) = \Theta(f(n))$, $f(n) = O(f(n))$, $f(n) = \Omega(f(n))$

**Symmetry:** $f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n))$

## Critical Limits

**Exp vs Poly:** $\lim_{n\to\infty} \frac{n^b}{a^n} = 0$ $(a > 1)$ $\implies n^b = o(a^n)$

**Poly vs Log:** $\lim_{n\to\infty} \frac{\lg^b n}{n^a} = 0$ $(a > 0)$ $\implies \lg^b n = o(n^a)$

**Factorial:** $n! = \omega(2^n)$, $n! = o(n^n)$

## Logarithms

$\lg n = \log_2 n$, $\ln n = \log_e n$

**Identities:**

$$a = b^{\log_b a}$$
$$\log(ab) = \log a + \log b$$
$$\log(a^k) = k \log a$$
$$\log_b a = \frac{\log a}{\log b}$$
$$\log(1/a) = -\log a$$
$$\log_b a = \frac{1}{\log_a b}$$

**Base irrelevant for Big-O:**

$\log_a n = \Theta(\log_b n)$ for any $a, b > 1$

## Exponentials

**Identities:** $a^0 = 1$, $a^1 = a$, $a^{-1} = 1/a$

$(a^m)^n = a^{mn}$, $a^m a^n = a^{m+n}$

**Natural exp:** $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$

$e^x \geq 1 + x$

$e^x = 1 + x + \Theta(x^2)$ as $x \to 0$

## Factorials

$n! = 1 \cdot 2 \cdot 3 \cdots n$

**Weak bound:** $n! \leq n^n$

**Stirling:** $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$

**Log factorial:** $\lg(n!) = \Theta(n \lg n)$

## Floor/Ceiling

$\lfloor x \rfloor$ = greatest int $\leq x$

$\lceil x \rceil$ = least int $\geq x$

$x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$

$\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$

## Modular Arithmetic

$a \bmod n = a - n\lfloor a/n \rfloor$

$0 \leq a \bmod n < n$

$a \equiv b \pmod{n} \iff (a \bmod n) = (b \bmod n)$

$\iff n | (b - a)$

## Fibonacci Numbers

$F_0 = 0$, $F_1 = 1$, $F_i = F_{i-1} + F_{i-2}$
**Golden ratio:** $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$
$\hat{\phi} = \frac{1-\sqrt{5}}{2} \approx -0.618$
$F_i = \frac{\phi^i - \hat{\phi}^i}{\sqrt{5}} \approx \frac{\phi^i}{\sqrt{5}}$
Fibonacci grows exponentially

## Iterated Logarithm

$\lg^* n = \min\{i \geq 0 : \lg^{(i)} n \leq 1\}$

| $n$ | $\lg^* n$ |
|---|---|
| 2 | 1 |
| 4 | 2 |
| 16 | 3 |
| 65536 | 4 |
| $2^{65536}$ | 5 |

Very slow growth!

## Loop Analysis

**Single loop:** for i=1 to n $\rightarrow O(n)$
**Nested (both n):** for i=1 to n, for j=1 to n $\rightarrow O(n^2)$
**Dependent:** for i=1 to n, for j=1 to i $\rightarrow \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = O(n^2)$
**Halving:** while n¿1: n=n/2 $\rightarrow O(\lg n)$
**Tree recursion:** T(n)=2T(n/2)+O(n) $\rightarrow O(n \lg n)$

## Summation Formulas

$\sum_{i=1}^{n} 1 = n$
$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2)$
$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \Theta(n^3)$
$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ $(a \neq 1)$
For $|x| < 1$: $\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$

## Practical Limits

*Assuming* $\sim 10^8$ *ops/sec:*

| **Complexity** | **Max** $n$ |
|---|---|
| $O(n)$ | $10^8$ |
| $O(n \lg n)$ | $10^7$ |
| $O(n^2)$ | $10^4$ |
| $O(n^3)$ | 500 |
| $O(2^n)$ | 25 |
| $O(n!)$ | 11 |

## Quick Rules

**Polynomials:** Drop lower terms & coefficients
$3n^2 + 5n + 2 = \Theta(n^2)$
**Nested loops:** Multiply complexities
2 nested n-loops $= O(n) \times O(n) = O(n^2)$
**Sequential:** Take maximum
$O(n) + O(n^2) = O(n^2)$
**Recursive:** Master Theorem or substitution

## Common Mistakes

"at least $O(n)$" (contradictory)
Confusing $2^n$ with $n^2$
Forgetting log base doesn't matter
Using $\Theta$ without proof
Counting input in space complexity

## Space Complexity

Count: aux data structures, recursion depth
Don't count: input size, output (unless total)
**Recursion depth d:** $O(d)$ stack space

## Master Theorem

For $T(n) = aT(n/b) + f(n)$:
Let $c_{crit} = \log_b a$
**Case 1:** $f(n) = O(n^c)$, $c < c_{crit}$
$\implies T(n) = \Theta(n^{c_{crit}})$
**Case 2:** $f(n) = \Theta(n^{c_{crit}})$
$\implies T(n) = \Theta(n^{c_{crit}} \lg n)$
**Case 3:** $f(n) = \Omega(n^c)$, $c > c_{crit}$
and regularity condition
$\implies T(n) = \Theta(f(n))$

## Example Analyses

| **Algorithm** | **Time** | **Space** |
|---|---|---|
| Linear search | $O(n)$ | $O(1)$ |
| Binary search | $O(\lg n)$ | $O(1)$ |
| Merge sort | $O(n \lg n)$ | $O(n)$ |
| Quick sort avg | $O(n \lg n)$ | $O(\lg n)$ |
| Quick sort worst | $O(n^2)$ | $O(n)$ |
| Heap sort | $O(n \lg n)$ | $O(1)$ |
| Bubble sort | $O(n^2)$ | $O(1)$ |
| Insertion sort | $O(n^2)$ | $O(1)$ |
| Hash lookup | $O(1)$ | $O(n)$ |
| BFS/DFS | $O(V + E)$ | $O(V)$ |

## Proving Bounds

**To prove $\Theta$:** Find $c_1, c_2, n_0$
OR prove both $O$ and $\Omega$
**To prove $O$:** Find $c, n_0$ for inequality
OR show $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$
**To prove $\Omega$:** Find $c, n_0$ for inequality
OR show $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$
**To disprove $O$:** Show $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$