的增长率。第三个定义 $T(N)=\Theta(h(N))$(念成“theta”)是说 $T(N)$的增长率等于 $h(N)$的增长率。最后一个定义 $T(N)=o(p(N))$(念成“小 o”)说的则是 $T(N)$的增长率小于 $p(N)$的增长率。它不同于大 O，因为大 O 包含增长率相同的可能性。

要证明某个函数 $T(N)=O(f(N))$，通常不是形式地使用这些定义，而是使用一些已知的结果。一般来说，这就意味着证明(或确定假设不成立)是非常简单的计算而不应涉及微积分，除非遇到特殊的情况(不可能在算法分析中发生)。

当 $T(N)=O(f(N))$时，我们是在保证函数 $T(N)$是在以不快于 $f(N)$的速度增长；因此 $f(N)$是 $T(N)$的一个**上界**(upper bound)。这意味着 $f(N)=\Omega(T(N))$，于是我们说 $T(N)$是 $f(N)$的一个**下界**(lower bound)。

作为一个例子，N^3 比 N^2 增长快，因此我们可以说 $N^2=O(N^3)$或 $N^3=\Omega(N^2)$。$f(N)=N^2$ 和 $g(N)=2N^2$ 以相同的速率增长，从而 $f(N)=O(g(N))$和 $f(N)=\Omega(g(N))$都是正确的。当两个函数以相同的速率增长时，是否需要使用记号 $\Theta(\ )$表示可能依赖于具体的上下文。直观地说，如果 $g(N)=2N^2$，那么 $g(N)=O(N^4)$，$g(N)=O(N^3)$和 $g(N)=O(N^2)$从技术上看都是成立的，但最后一个是最佳选择。写法 $g(N)=\Theta(N^2)$不仅表示 $g(N)=O(N^2)$而且还表示结果尽可能地好(严密)。

我们需要掌握的重要结论为：

法则 1：

如果 $T_1(N)=O(f(N))$且 $T_2(N)=O(g(N))$，那么

(a) $T_1(N)+T_2(N)=O(f(N)+g(N))$(直观地和非正式地可以写成 $\max(O(f(N)), O(g(N)))$)。

(b) $T_1(N)*T_2(N)=O(f(N)*g(N))$。

法则 2：

如果 $T(N)$是一个 k 次多项式，则 $T(N)=\Theta(N^k)$。

法则 3：

对任意常数 k，$\log^k N=O(N)$。它告诉我们对数增长得非常缓慢。

这些信息足以按照增长率对大部分常见的函数进行分类(见图 2-1)。

有几点需要注意。首先，将常数或低阶项放进大 O 是非常坏的习惯。不要写成 $T(N)=O(2N^2)$或 $T(N)=O(N^2+N)$。在这两种情形下，正确的形式是 $T(N)=O(N^2)$。这就是说，在需要大 O 表示的任何分析中，各种简化都是可能发生的。低阶项一般可以被忽略，而常数也可以弃掉。此时，要求的精度是很粗糙的。

函数	名称
c	常数
$\log N$	对数
$\log^2 N$	对数平方的
N	线性的
$N\log N$	
N^2	二次的
N^3	三次的
2^N	指数的

图 2-1 典型的增长率

第二，我们总能够通过计算极限 $\lim_{N\to\infty} f(N)/g(N)$来确定两个函数 $f(N)$和 $g(N)$的相对增长率，必要的时候可以使用洛必达法则[⊖]。该极限可以有四种可能的值：

- 极限是 0：这意味着 $f(N)=o(g(N))$。
- 极限是 $c\neq 0$：这意味着 $f(N)=\Theta(g(N))$。

⊖ 洛必达法则(L'Hôpital's rule)说的是，若 $\lim_{N\to\infty} f(N)=\infty$ 且 $\lim_{N\to\infty} g(N)=\infty$，则 $\lim_{N\to\infty} f(N)/g(N)=\lim_{N\to\infty} f'(N)/g'(N)$，而 $f'(N)$和 $g'(N)$分别是 $f(N)$和 $g(N)$的导数。