链滴

算法分析（一）：数学基础

作者：Angonger

原文链接：https://ld246.com/article/1500981789632

来源网站：链滴

定义

1. 如果存在正常数c和n ₀使得当N≥n ₀时T N)≤cf(N),则计为T(N)=O(f(N))。
2. 如果存在正常数c和n ₀使得当N≥n ₀时T N)≥cg(N),则计为T(N)=Ω(g(N))。
3. T(N)=θ(h(N))当且仅当T(N)=O(h(N))和T(N)=Ω(g(N))。
4. 任意正常数c都存在常数n ₀使得当N≥n ₀ T(N)<cp(N),则T(N)=o(p(N))。

看不懂没关系，算法主要用这些来比较**相对增长率**，定义中四个公式分别表示

1. T(N)的增长率 **小于或等于**f(N)的增长率，f(N)是T(N)的**上界（upper bound）**
2. T(N)的增长率 **大于或等于**g(N)的增长率，T(N)是f(N)的**下界（lower bound）**
3. T(N)的增长率 **等于**h(N)的增长率
4. T(N)的增长率 **小于**p(N)的增长率

eg:

N³比N²增长快，则可以表示为N²=O(N<sup>3</su>)

需要掌握的结论

1. 如果T ₁(N)=O(f(N))且T₂(N)=O(g(N)),那么 (a) T₁ N)+T₂(N)=O(f(N)+g(N));直观表达为max(O(f(N)),O(g(N)))。

(b) T₁(N)*T₂(N)=O(f(N)*g(N))。

2. 如果T(N)是一个k次多项式，则T(N)=θ(N ^k)
3. 对任意常数k，log ^kN=O(N);对数增长的特别缓慢。

注意

1. 尽量不要将常数或者低阶项放入大O中，因为算法分析中常数和低阶项很有可能被忽略。
2. 总能通过计算极限lim _{N→∞}f(N)/g(N)来确定两个函数的相对增长率，必要可使洛必达法则。有四种可能值

2.1 极限为0：f(N)=o(g(N))。

2.2 极限是c≠0：f(N)=θ(g(N))。

2.3 极限是∞：g(N)=o(o(f(N)))。

2.4 极限摆动：二者无关。

参考资料《数据结构与算法分析》

原文链接：算法分析（一）：数学基础