浅谈关于时间复杂度的几个渐进记号[1]

一. 前置知识

a) T(n)不是一个记号，而是最坏情况下的运行时间（计算量）的函数

b) 渐进：众所周知，我们在计算复杂度时是忽略了低次项及常数的，所以在自变量较小的情况下，可能不满足渐进符号给出的上下界。因此我们引入渐进，即随着自变量的增加函数逐渐符合渐进符号给出的上下界。（但不是逐渐接近）[2]

二. 常用记号的文字定义

a) Θ记号：函数的渐进上下界

b) O 记号：函数的渐进上界

c) Ω记号：函数的渐进下界

d) o 记号：函数的非渐进紧确的上界

e) ω记号：函数的非渐进紧确下界

三. 常用记号的形象描述

a) Θ记号：=

b) O 记号：⩽

c) Ω记号：⩾

[1] 本文提到的记号全部适用于函数，但本文仅就其在描述复杂度方面的应用进行讨论

[2] 关于渐进的确切定义，在网络上和《算法导论》上均没有找到，故这里的理解是自己的理解，肯能有错，欢迎指正。

d) o 记号：<

e) ω 记号：>

四. 常用记号的图像描述[3]

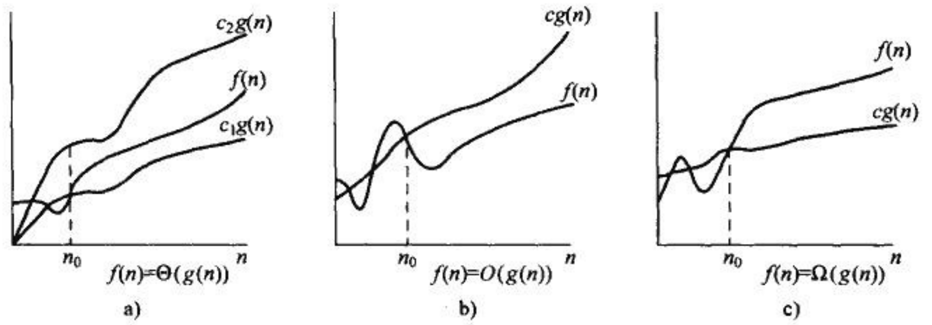

图 3-1 记号 Θ，O 和 Ω 的图例。在每个部分中，n_0 是最小的可能值；大于 n_0 的值也有效。a)Θ 记号限制一个函数在常数因子内。如果存在正常数 n_0，c_1 和 c_2 使得在 n_0 右边 $f(n)$ 的值永远在 $c_1g(n)$ 与 $c_2g(n)$ 之间，那么可以写成 $f(n)=\Theta(g(n))$。b)O 记号给出一个函数在常数因子内的上限。如果存在正常数 n_0 和 c 使得在 n_0 右边 $f(n)$ 的值永远等于或小于 $cg(n)$，那么可以写成 $f(n)=O(g(n))$。c)Ω 符号给出一个函数在常数因子内的下限。如果存在正常数 n_0 和 c 使得在 n_0 右边 $f(n)$ 的值永远等于或大于 $cg(n)$，那么可以写成 $f(n)=\Omega(g(n))$

五. Reference：

a） 《算法导论》第一部分·第三章

b） 《渐近记号总结》

[3] 摘自算法导论