在渐进分析中等于符号的滥用

我们用$f(n) = O(g(n))$来表示 $f(n)$ 是 $O(g(n))$的一个成员函数而不用传统的$T(n) \in O(f(n))$来表示。原因？没有原因，习惯问题而已。

- 例子
  - $f(n) = 5n^3$; $g(n) = 3n^2$
  - $f(n) = O(n^3) = g(n)$
  - 但是 $f(n) \neq g(n)$.
- 更好的表式方式: $T(n) \in O(f(n))$.
  - $5n^3 \in O(n^3)$
  - $3n^2 \in O(n^2) \subset O(n^3)$

算法复杂性分析

例如：$N>=1, 3N<=4N$, 有$3N=O(N)$

$N>=1, N+1024<=1025N$,有$N+1024=0（N)$

$N>=10$，$2N^2+11N-10<=3N^2, 2N^2+11N-10=0(N^2)$

$N>=1, N^2<N^3, N^2=O(N^3)$

证明：$N^3 \neq O(N^2)$ 。

假设存在正的常数C和自然数N_0，使得当$N>=N_0$时有$N^3<=CN^2$。显然，当取$N=\max\{N_0, \lfloor C \rfloor+1\}$时不等式不成立，所以得证

再次分析

$\cdot 100n^2=1.5n^2=n^2+4=10n^2-3n+6=\Theta(n^2)$.

课堂讨论

- 根据所学的渐进分析方法，同学们分析下下面的结论是否正确

例如：　$f(n) = 32n^2 + 17n + 32$.

$f(n)$ 属于 $O(n^2)$, $O(n^3)$, $\Omega(n^2)$, $\Omega(n)$, $\Theta(n)$.

$f(n)$ 不属于 $O(n)$, $\Omega(n^3)$, $\Theta(n^2)$, or $\Theta(n^3)$.

渐进分析的算术运算：

- $O(f(n))+O(g(n)) = O(\max\{f(n),g(n)\})$ ；
- $O(f(n))+O(g(n)) = O(f(n)+g(n))$ ；
- $O(f(n))*O(g(n)) = O(f(n)*g(n))$ ；
- $O(cf(n)) = O(f(n))$ ；
- $g(n)= O(f(n)) \Rightarrow O(f(n))+O(g(n)) = O(f(n))$ 。

- 规则$O(f(n))+O(g(n)) = O(\max\{f(n),g(n)\})$ 的证明:
- 对于任意$f_1(n) \in O(f(n))$，存在正常数c_1和自然数n_1，使得对所有$n \geq n_1$，有 $f_1(n) \leq c_1 f(n)$ 。
- 类似地，对于任意$g_1(n) \in O(g(n))$，存在正常数c_2和自然数n_2，使得对所有 $n \geq n_2$，有$g_1(n) \leq c_2 g(n)$ 。
- 令$c_3=\max\{c_1, c_2\}$， $n_3 =\max\{n_1, n_2\}$，$h(n)= \max\{f(n),g(n)\}$ 。
- 则对所有的 $n \geq n_3$，有
- $f_1(n) + g_1(n) \leq c_1 f(n) + c_2 g(n)$
- $\leq c_3 f(n) + c_3 g(n)= c_3(f(n) + g(n))$
- $\leq c_3 2 \max\{f(n),g(n)\}$
- $= 2c_3 h(n) = O(\max\{f(n),g(n)\})$.