Algorithm 0x00

49.234.77.58/index.php/2019/12/15/algorithm-0x00

XZLang

2019年12月15日

O（读“O”）：上界

Ω（读”omega”）：下界

Θ（读”theta”）：近似

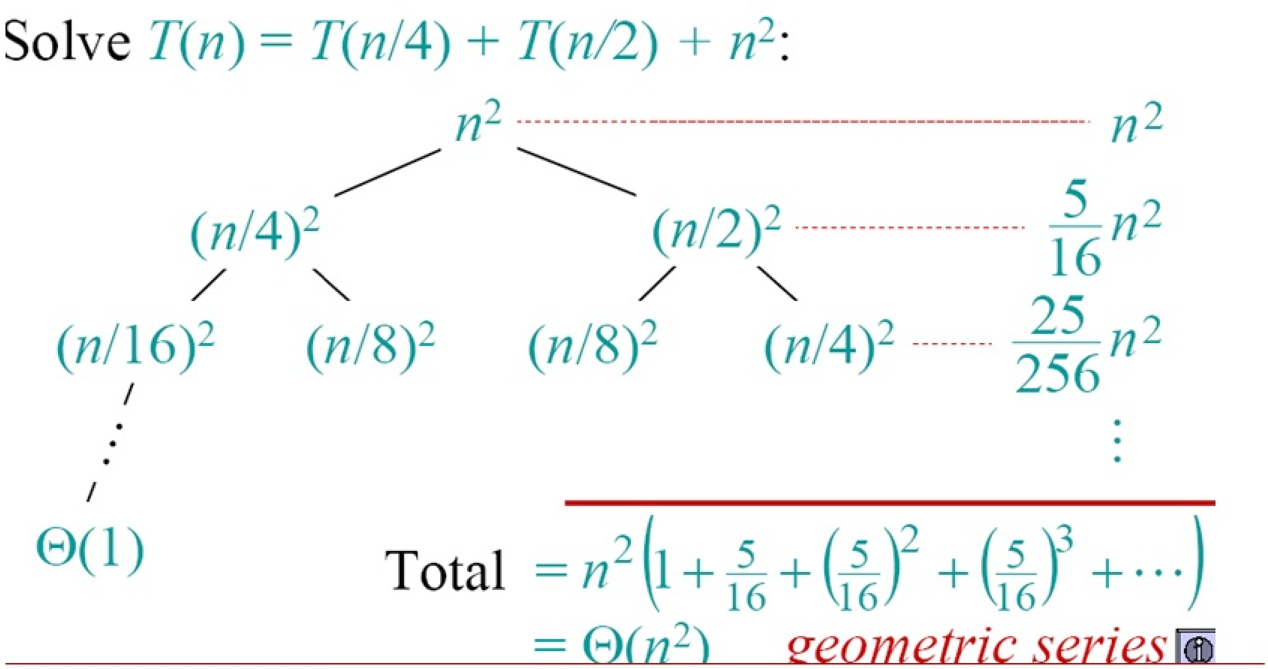
对于足够大的*n*，*t(n)*的上界由*g(n)*的常数倍来确定，即：
t(n)<=Cg(n)，记作t(n)=O(g(n))

当且仅当存在正的常数**C1**和**C2**，使得对于所有的 **$n>=n_0$**，有 **C1g(n)≤ f(n) ≤ C2g(n)**，
记作： **f(n)= Θ(g(n))**

$$
\begin{aligned}
T(n) &= 4T(n/2) + n \\
&\le 4c(n/2)^3 + n \\
&= (c/2)n^3 + n \\
&= cn^3 - ((c/2)n^3 - n) \leftarrow \textit{desired} - \textit{residual} \\
&\le cn^3 \leftarrow \textit{desired}
\end{aligned}
$$

whenever $(c/2)n^3 - n \ge 0$, for example,
if $c \ge 2$ and $n \ge 1$.

residual



主定理

定理：设$a\geq 1$, $b>1$为常数, $f(n)$为函数, $T(n)$为非负整数，且$T(n)=aT(n/b)+f(n)$，则

1. 若 $f(n)=O(n^{\log_b a-\varepsilon})$, $\varepsilon>0$, 那么

   $T(n)=\Theta(n^{\log_b a})$

   存在ε

2. 若 $f(n)=\Theta(n^{\log_b a})$, 那么

   $T(n)=\Theta(n^{\log_b a}\log n)$

   存在ε

3. 若 $f(n)=\Omega(n^{\log_b a+\varepsilon})$, $\varepsilon>0$,且对于某个常数 $c<1$和充分大的 n 有$af(n/b)\leq cf(n)$, 那么

   $T(n)=\Theta(f(n))$

   存在c和n_0