时间复杂度

Time complexity of Insertion Sort

Insertion-Sort (A)	cost	# of times
`for j = 2 to A.length`	c_1	n
`  key = A[j]`	c_2	$n-1$
`  i = j - 1`	c_3	$n-1$
`  while (i>0 and A[i]>key)`	c_4	$\sum_{j=2}^{n} t_j$
`    A[i+1] = A[i]`	c_5	$\sum_{j=2}^{n} (t_j - 1)$
`    i = i - 1`	c_6	$\sum_{j=2}^{n} (t_j - 1)$
`  A[i+1] = key`	c_7	$n-1$

Assume $A.length = n$, then total running time $T(n)$ is:

$$c \cdot n + c' \cdot \left(\sum_{j=2}^{n} t_j\right) - c''$$

If $t_j = 1$, then $T(n) \approx cn + c'n - c''$ **Best case:** A is sorted

If $t_j = j$, then $T(n) \approx cn + (c'/2)n^2 - c''$ **Worst case:** A is reversely sorted

Average case???

$$T(n) = O(n^2)$$

Runtime $T(n) = c \cdot n + c' \cdot \left(\sum_{j=2}^{n} t_j\right) - c''$

Best case: $T(n) = \Theta(n)$

$t_j = 1$ and $T(n) \approx cn + c'n - c''$

Worst case: $T(n) = \Theta(n^2)$

$t_j = j$ and $T(n) \approx cn + (c'/2)n^2 - c''$

5记号

O：与ω逻辑反，渐进意义上的线性上界，算法很好

Ω：与O意义相反，与o逻辑反，代表算法时间下界，算法不太好

Θ：区间

o：任意系数c

ω：任意系数c

- 有自反性，对称性

渐进记号比较

- 洛必达
- 斯特勒公式：估计阶乘