

排序

快速排序：比较次数

14-A5

时间究竟是什么

假使人家不问我，我像很明了

假使要我解释起来，我就茫无头绪

邓俊辉

deng@tsinghua.edu.cn

递推分析 (1/2)



❖ 记期望的比较次数为 $T(n)$: $T(1) = 0, T(2) = 1, \dots$

❖ 可以证明 : $T(n) = \mathcal{O}(n \log n) \dots$

$$T(n) = (n-1) + \frac{1}{n} \cdot \sum_{k=0}^{n-1} [T(k) + T(n-k-1)] = (n-1) + \frac{2}{n} \cdot \sum_{k=0}^{n-1} T(k)$$

$$n \cdot T(n) = n \cdot (n-1) + 2 \times \sum_{k=0}^{n-1} T(k)$$

$$(n-1) \cdot T(n-1) = (n-1) \cdot (n-2) + 2 \times \sum_{k=0}^{n-2} T(k)$$

递推分析 (2/2)



$$n \cdot T(n) - (n-1) \cdot T(n-1) = 2 \cdot (n-1) + 2 \times T(n-1)$$

$$n \cdot T(n) - (n+1) \cdot T(n-1) = 2 \cdot (n-1)$$

$$\frac{T(n)}{n+1} - \frac{T(n-1)}{n} = \frac{4}{n+1} - \frac{2}{n}$$

$$\frac{T(n)}{n+1} = \frac{T(n)}{n+1} - \frac{T(1)}{2} = 4 \cdot \sum_{k=2}^n \frac{1}{k+1} - 2 \cdot \sum_{k=2}^n \frac{1}{k} = 2 \cdot \sum_{k=1}^{n+1} \frac{1}{k} + \frac{2}{n+1} - 4 \approx 2 \cdot \ln n$$

$$T(n) \approx 2 \cdot n \cdot \ln n = (2 \cdot \ln 2) \cdot n \log n \approx 1.386 \cdot n \log n$$

后向分析

❖ 设经排序后得到的输出序列为： $\{ a_0, a_1, a_2, \dots, a_i, \dots, a_j, \dots, a_{n-1} \}$

这一输出结果与具体使用何种算法无关，故可使用 Backward Analysis

❖ 比较操作的期望次数，应等于

每一对元素 $\langle a_i, a_j \rangle$ 在排序过程中接受比较之概率的总和：

$$T(n) = \sum_{j=1}^{n-1} \sum_{i=0}^{j-1} Pr(i, j)$$

❖ quickSort() 的计算过程及实质结果，可理解为：按某种次序，逐个将各元素确认为 pivot

1. 若 $k \in [0, i) \cup (j, n)$ ，则 a_k 早于或晚于 a_i 和 a_j 成为 pivot，与 $Pr(i, j)$ 无关

2. 实际上： $\langle a_i, a_j \rangle$ 接受比较，当且仅当

在 $\{ a_i, a_{i+1}, a_{i+2}, \dots, a_{j-2}, a_{j-1}, a_j \}$ 中， a_i 或 a_j 率先被确认

$$T(n) = \sum_{j=1}^{n-1} \sum_{i=0}^{j-1} Pr(i, j) = \sum_{j=1}^{n-1} \sum_{d=1}^j \frac{2}{d+1} \approx \sum_{j=1}^{n-1} 2 \cdot (\ln j - 1) \leq 2 \cdot n \cdot \ln n$$

对比

	#compare	#move (对实际性能影响更大)
Quicksort	平均 $O(1.386*n\log n)$ 且高概率接近	平均不超过 $O(1.386*n\log n)$ 且实际更少
Mergesort	严格 $O(1.00*n\log n)$	严格 $O(1.00*n\log n)$ 实际往往加倍