# Day4 排序进阶 · 课堂笔记

## 01 随机快速排序

$n \le 1$：直接返回。随机取枢轴 $x$。
分：先取出 $x$，其余按 $\le x$ / $> x$ 分成 $B$、$C$。
治：$QuickSort(B)$，$QuickSort(C)$。
合：$B + [x] + C$（枢轴不再参与递归）。

---

$Partition(A[p..r])$：先把枢轴换到 $A[r]$。
$x = A[r]$，$i = p-1$；扫描 $j = p, \cdots, r-1$。
若 $A[j] \le x$：$i$++，交换 $A[i]$、$A[j]$。
最后交换 $A[i+1]$，$A[r]$，返回 $i+1$。
不变式：$A[p..i] \le x$；一趟 $\Theta(n)$。

例：$x = 20$

| 40 | 30 | 10 | 20 | → | 10 | 30 | 40 | 20 | → | 10 | 20 | 40 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## 02 运行时间看怎么分

接近一半：

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) \Rightarrow O(n \log n)$$

每次分成 0 与 $n-1$：

$$T(n) = T(n-1) + \Theta(n) \Rightarrow O(n^2)$$

再加一步：先检查是否有序，有序就返回。
最好输入：$O(n)$，检查完即返回。
困难输入，随机取得最好：$O(n \log n)$。
困难输入，最坏随机选择：$O(n^2)$。

## 03 期望时间怎么来？

先按元素互异分析；固定输入，对随机枢轴取期望。
枢轴的秩均匀：
$(|B|, |C|) = (0, n-1), (1, n-2), \cdots, (n-1, 0)$，各为 $\frac{1}{n}$。
把一趟线性代价简记为 $n$，$T(n)$ 表示期望时间：

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

$$= \frac{2}{n}\sum_{i=0}^{n-1} T(i) + n$$

$$nT(n) = 2\sum_{i=0}^{n-1} T(i) + n^2$$

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

相减：$nT(n) = (n+1)T(n-1) + 2n - 1$

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

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

展开：$\frac{T(n)}{n+1} \le \frac{T(1)}{2} + 2\sum_{j=2}^{n}\frac{1}{j+1} = O(\log n)$

于是 $T(n) = O(n \log n)$。

旁注：Quickselect 找中位数，期望 $O(n)$（课件此处应为 $O(n)$）。

## 04 堆排序

能否每次更快地取出最小值？
用最小堆 min-heap。
从 $n$ 个数建堆：$O(n)$。
堆顶就是最小值：$O(1)$。
删除堆顶并恢复堆：$O(\log n)$。
重复 $n$ 次，依次输出最小值。
总时间 $O(n + n \log n) = O(n \log n)$。

这些都基于比较。
元素不必是数字，只要能相互比较。

## 05 计数排序

若输入全是正整数，值域为 $1, \cdots, k$。
例：$A = [1, 5, 3, 2, 2]$。
① 扫一遍 $A$，找最大值 $k$。
② $C[1..k]$ 置 0；对每个 $x$：$C[x]$++。
③ 扫 $C$，按序输出 $C[x]$ 个 $x$。
时间 $O(n+k)$；空间 $O(n+k)$（含输出）。
这是多项式时间吗？
要看二进制输入长度，而非数值大小。
每个数约 $\log_2 k$ 位，总长约 $n \log_2 k$。
$k$ 可相对 $\log k$ 指数大。
所以 $O(n+k)$ 是伪多项式时间。
不是正整数的情形，课件接着提到桶排序。

## 06 比较排序还能更快吗？

一次比较只有两种结果，算法可以表示为二叉决策树。
对 $n$ 个互异元素，有 $n!$ 种可能的次序。
每个叶子对应一种次序；每种次序至少要有一个叶子。
若最坏比较次数为 $h$，树高为 $h$，叶子数最多 $2^h$。
$2^h \ge n! \Rightarrow h \ge \log_2(n!) = \Theta(n \log n)$。
因此，任何基于比较的排序，最坏情况都需 $\Omega(n \log n)$ 次比较。

