

# Notes for Tutorial Three

## 一、序关系：从排序到选择 【Ref: 算法导论】

结合书中定义理解的序关系  $\begin{cases} \text{全序, 若元素之间两两都有关系} \\ \text{偏序, 若元素之间只部分有关系} \end{cases}$

从序关系的角度审视排序算法：快速排序与归并排序的对偶关系

从序关系的角度审视 Rank(k)的意义：借用快速排序的分割的办法

### 例 1: 设计找出序列中前 k 大的元素，且找出的元素需有序排列

- (1) 求  $O(n \log n)$  的算法 -> 直接全部排序，找出前 k 个元素；
- (2) 求  $O(n + k \log n)$  的算法 -> 建立最小堆，每次取最小元素后 fix，取 k 次；
- (3) 求  $O(n + k \log k)$  的算法 -> 先选出前 k 小的元素，再对它们排序；

延伸思考：

**武器库理论**：将已知的算法按照不同的维度归类整理，解题遇到相关信息时从对应的库中检索“武器”。如：在  $O(n)$  时间内能完成的事：选择、Partition、建堆等；

拓展说明：

**算法设计类题目的答题格式：**

答案中涉及的原理和概念+算法的简要文字说明+详细的算法设计(Step1, Step2, ...);

## 二、关于平衡性的理解

### 例 2: 求平面中的 Maxima 点的问题

列出递归式： $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$   $\begin{cases} \text{Select Median} \\ \text{Partition} \\ \text{合并} \end{cases}$

此题的下界是  $\Omega(n \log n)$

### 例 3: “改进”版的 Quick Sort

在  $O(n)$  时间内我们可以做三件事： $\begin{cases} \text{Select Median} \\ \text{用找到的 Median 作为 pivot} \\ \text{划分和解决子问题} \end{cases}$

分析：由于 Worst case selection 的限制，此算法的性能极差，不如随机选取 pivot 的快速排序算法

### 例 4: 芯片测试问题 【Ref: 算法导论】

如何找出一个好芯片来判断所有芯片的好坏

由第 (1) 问可知：用 PK 法可解此题，取芯片两两比较，若判定为可能的后三种情况，则将两个芯片都扔掉，若判定为两片都为好芯片或都为坏芯片的情况，则任取一片扔掉，剩下的一片进入下一轮比较。

### 三、 Partition 和 Selection 的化学作用

#### 例 5: 找出离 Median 最近的 k 个数

算法的本质在于找出离 median 有 k 个距离以内的数

#### 例 6: Weighted Median 问题

- (1) 当权重为 $\frac{1}{n}$ 时, 退化为传统的 median 问题;
- (2)  $O(n \log n)$ 的算法 -> 对所有元素排序后, 从头遍历并累加权值统计;
- (3)  $O(n)$ 的算法 -> 先找出传统 median, 根据它进行划分, 扫描并累加左右的权值, 把不需要的那一半元素扔掉, 并把它们的权值之和赋给中位数, 继续下一次划分。

$$\text{复杂度: } n + \frac{n}{2} + \frac{n}{4} + \cdots = O(n)$$

### 四、 查找

#### 例 7: 计算根号 N 【Ref: 习题课】

- (1) 假设对 n 个比特的数进行移位和加法的时间复杂度为 1 个时间单位;
- (2) 利用上次折半的结果来计算下一次折半的值;