Tutorial 3

balance：更高级的分治

初级平衡性

最坏情况线性时间选择

元素分组，median 的 median。最坏情况下的平衡性得到了保证

Balanced-QuickSort

先以 $O(n)$ select-median，再用 median 做 pivot 进行 partition

问题：面向实际时代价过大

芯片测试问题

> CRLS 3rd edition P.109
>
> 4-5 Chip testing Professor Diogenes has n supposedly identical integrated-circuit chips that in principle are capable of testing each other. The professor's test jig accommodates two chips at a time. When the jig is loaded, each chip tests the other and reports whether it is good or bad. A good chip always reports accurately whether the other chip is good or bad, but the professor cannot trust the answer of a bad chip. Thus, the four possible outcomes of a test are as follows:
>
> | Chip A says | Chip B says | Conclusion |
> | --- | --- | --- |
> | B is good | A is good | both are good, or both are bad |
> | B is good | A is bad | at least one is bad |
> | B is bad | A is good | at least one is bad |
> | B is bad | A is bad | at least one is bad |

> 1. Show that if more than $n/2$ chips are bad, the professor cannot necessarily determine which chips are good using any strategy based on this kind of pairwise test. Assume that the bad chips can conspire to fool the professor.
> 2. Consider the problem of finding a single good chip from among n chips, assuming that more than $n/2$ of the chips are good. Show that $\lfloor n/2 \rfloor$ pairwise tests are sufficient to reduce the problem to one of nearly half the size.
> 3. Show that the good chips can be identified with $\Theta(n)$ pairwise tests, assuming that more than $n/2$ of the chips are good. Give and solve the recurrence that describes the number of tests.

找出所有好芯片（假设好芯片个数大于坏芯片：坏芯片如果严格大于好芯片，不存在任何算法能分辨出所有好芯片：adversary argument）

目标：找出一块好的芯片，与其余芯片比较

思路：两两分组，只要不是 case 1 均去掉（去掉至少一个坏芯片，去掉之后好芯片仍然比坏芯片多，*"好人总比坏人多，牺牲至多一个好人干掉至少一个坏人"*）

adversary strategy：全部都是 case 1，则直接去掉每组其中一个

可以每次去掉一半芯片，最终得出一个好芯片，以 $O(n)$ 时间与其他芯片比较

计算 $\sqrt(N)$

critical operation：n bit 数的移位和加法

整数相乘代价：移位 n 次相加 n 次，代价为 $O(n)$

本质：查找问题，折半查找

最多递归 $\log N = n$ 次，代价为 $n \cdot O(n) = O(n^2)$

有没有可能把代价降到 $O(n)$？

利用之前的结果，假设计算出 $\left(\frac{1}{2}N\right)^2 = \frac{1}{4}N^2$

- 向左递归 $\left(\frac{1}{4}N\right)^2 = \frac{1}{16}N^2 = \frac{1}{4}N^2 \times \frac{1}{4}$，移位两次
- 向右递归 $\left(\frac{3}{4}N\right)^2 = \frac{9}{16}N^2 = \frac{1}{4}N^2 \times \frac{1}{4} + \frac{1}{4}N^2 \times 2$，移位三次加法一次

代价为 $n \cdot O(1) = O(n)$

精确平衡性

Maxima 回顾

划分一次：用 median 划分（不考虑 worst-case 性能，可随机选择 pivot）

划分两次：可能达到，但性能不止 $O(n)$ ，单纯选择横坐标和纵坐标的 median 不能四等分

Partition vs Selection

> 矬矬的 partition -> 矬矬的 selection -> 加强的 partition -> 加强的 selection
>
> “像人间大炮一级准备，二级准备一样”
>
> 分布式

找前 k 大的元素

假设 $k \ll n$

- $n \log n$ ：排序
- $n + k \log n$ ：建堆 + k 次修复
- $n + k^2$ ：建堆 + k 次部分修复（仅修复前 k 层）
- $n + k \log k$ ：select k^{th} + 排序

寻找离 median 最近的 k 个数

思路：select median 找出 median，然后 select $n/2 - k$ 和 $n/2 + k$，作为 pivot 划分

或是排序

weighted median

每个元素 x_i 有权重 $w_i, \sum_{i=1}^{n} w_i = 1$

若 x_k 满足

$$\sum_{i<k} w_i < \frac{1}{2}$$

$$\sum_{i>k} w_i \leqslant \frac{1}{2}$$

则称 x_k 为 weighted median

weighted median 是普通 median 的泛化

> 引理：若 $w_i = \dfrac{1}{n}$，则一个元素是 weighted median $\iff$ 它是 median

- $O(n \log n)$：排序后扫描
- $O(n)$：select median（线性时间）后 partition，检查权之和，若不符合，将其中一半的权之和加在 median 上，删除那部分元素，再在另一半递归求解

分析策略

Amortized analysis：湖景房问题

湖边有 n 个房子 $\{a_1, a_2, \ldots, a_n\}$，给出高度，求问有多少湖景房

算法：一个栈，有元素来时，和栈内所有元素比较

- 如果比栈内元素高，则出栈
- 直到没有更高的，入栈

循环不变式：栈中的元素越来越矮，归纳法可证

amortized analysis：最终代价为 $O(n)$

Adversary argument：找出唯一不同元素

所有元素中除一个元素以外其余均一样，求唯一不同的元素，critical operation 是比较是否相同

adversary strategy：每次比较，令两个元素一样

- 剩三个元素：取两个元素比较，令其不一样，再比较一次
- 剩四个元素：令每次比较的元素一样，比较两次

比较次数下界：$\left\lceil \dfrac{n}{2} \right\rceil$

adversary argument 非常强大