

计算机科学导论

张家琳

中国科学院计算技术研究所

zhangjialin@ict.ac.cn

2025-4-18

思考题

$$\frac{1}{2^n}$$

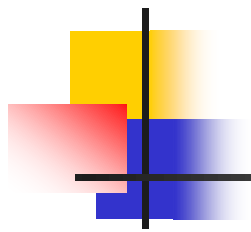
- 问题： n 名同学围成一圈，每个人随机的分配一顶红色或者蓝色的帽子，要求所有人同时猜出自己帽子的颜色。请设计一种策略，使得同时猜对的概率最高。

$\frac{1}{2}$
假设 $x_i = 1$ 表示第 i 个人带红帽子，
 $= 0$ 表示第 i 个人带蓝帽子

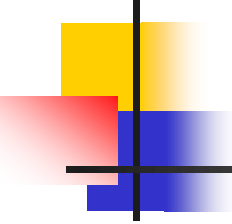
策略：猜测

$$x_1 \oplus x_2 \oplus \dots \oplus x_n = 0$$





算法思维



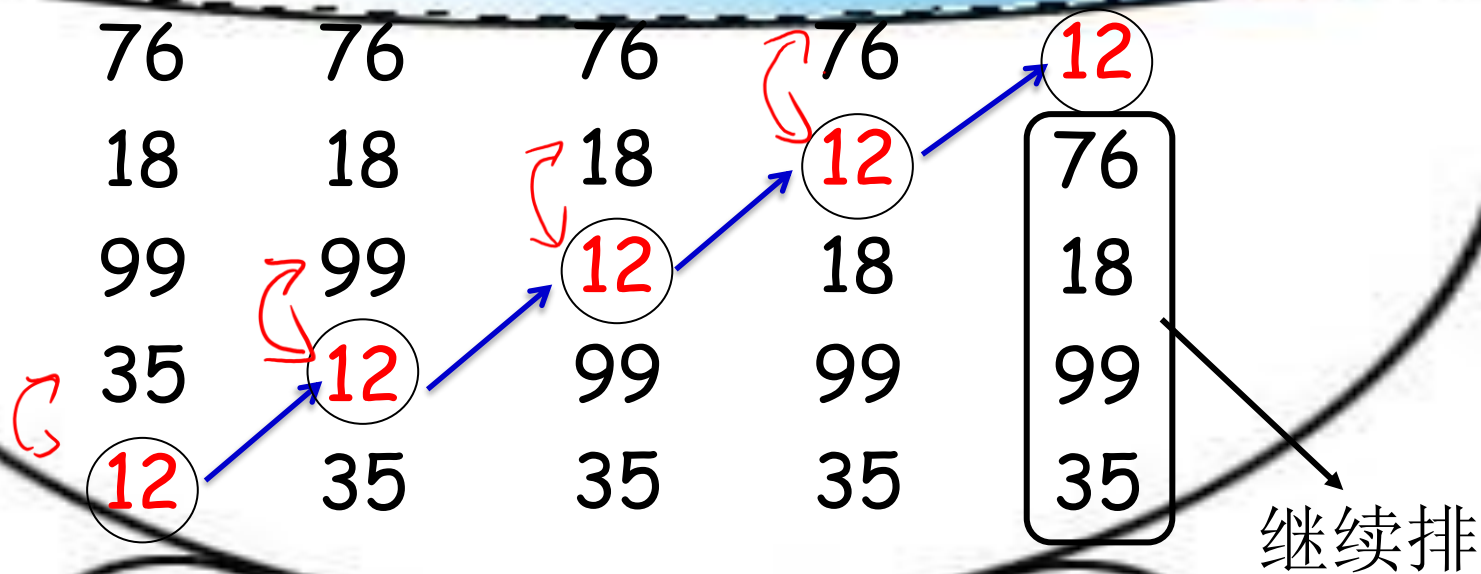
引入：排序

32, 25, 17, 64, 81

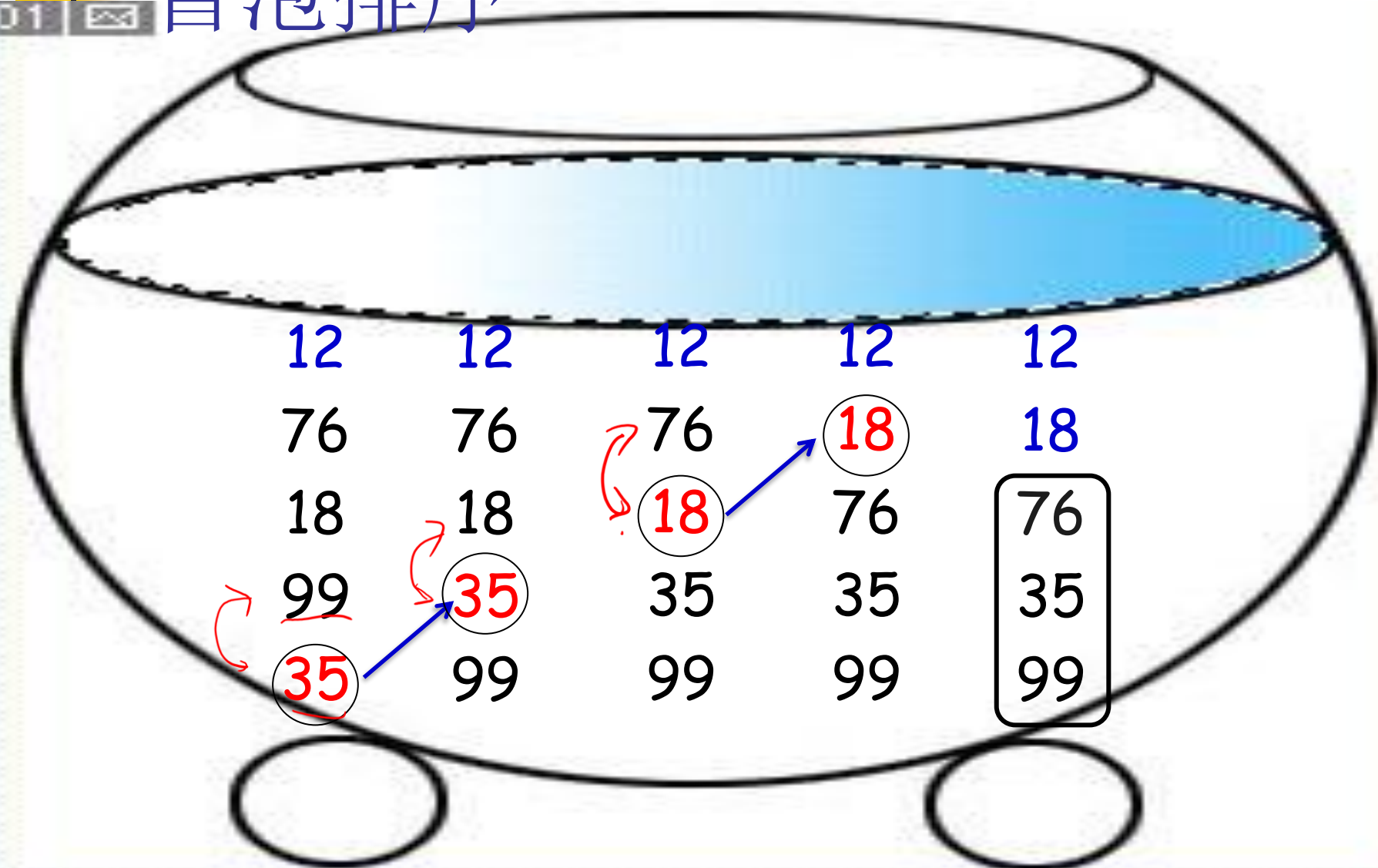
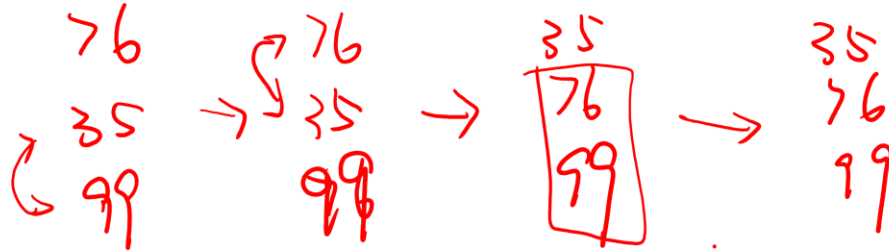
17, 25, 32, 64, 81

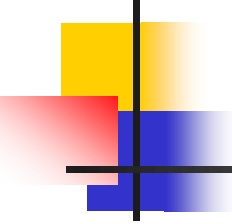
- 任务：给定n个正整数，把他们从小到大排起来
- 请思考在斗地主、升级、掼蛋.....的时候，你是怎么整理牌的？
 - 一张一张摸牌，每次插入已经理好的牌里面（插入排序法）
 - 把最大的牌放到最左边（选择排序法）
 - 先按花色分，再各个花色整理

冒泡排序



冒泡排序



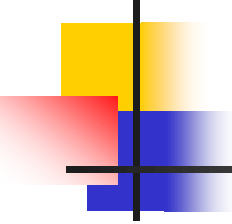


算法

BBB 15, 21, 5, 37, BBB

BBB 5, 15, 21, 37, BBB

- 有穷性。
 - 确切性。
 - 输入。
 - 输出。
 - 可行性。
-
- 例子：把n个数中的最小数放到第一个。



冒泡排序

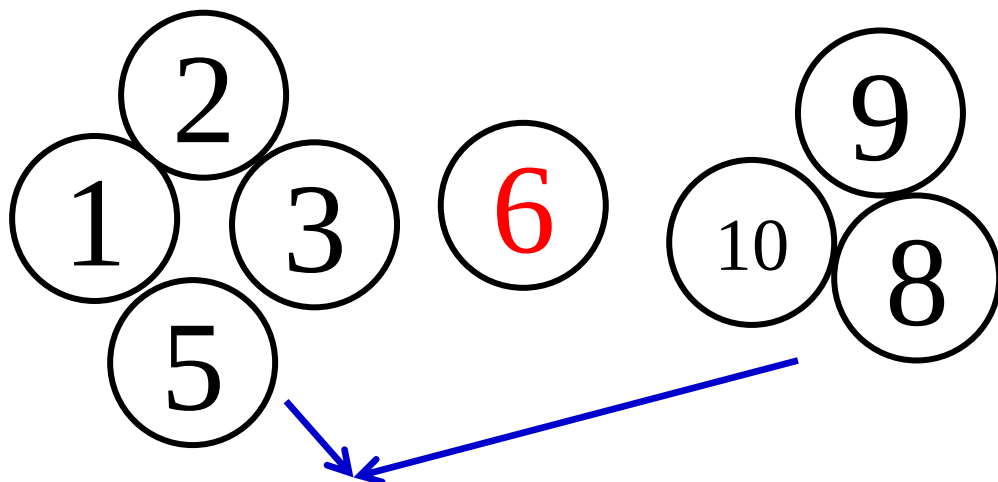
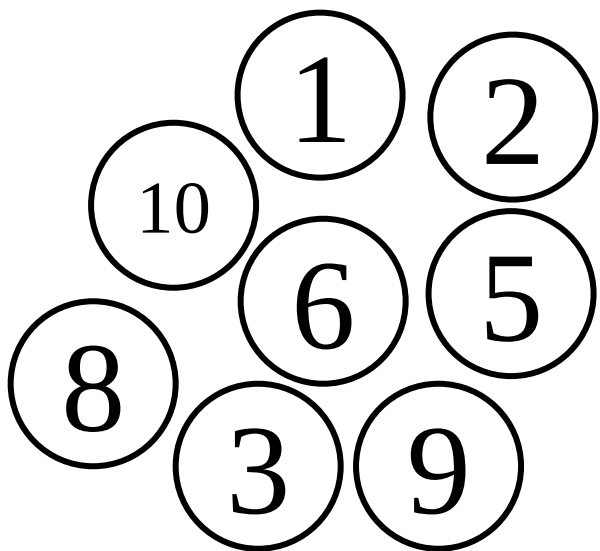
- 需要 $n-1+n-2+\dots+1=n \times (n-1)/2$ 次比较操作
- 最坏情况下需要 $n(n-1)/2$ 次交换操作
 - $n, n-1, \dots, 2, 1$
- 能不能更快？

快速排序

伪随机数发生器

Step1: 随机选其中一个数字

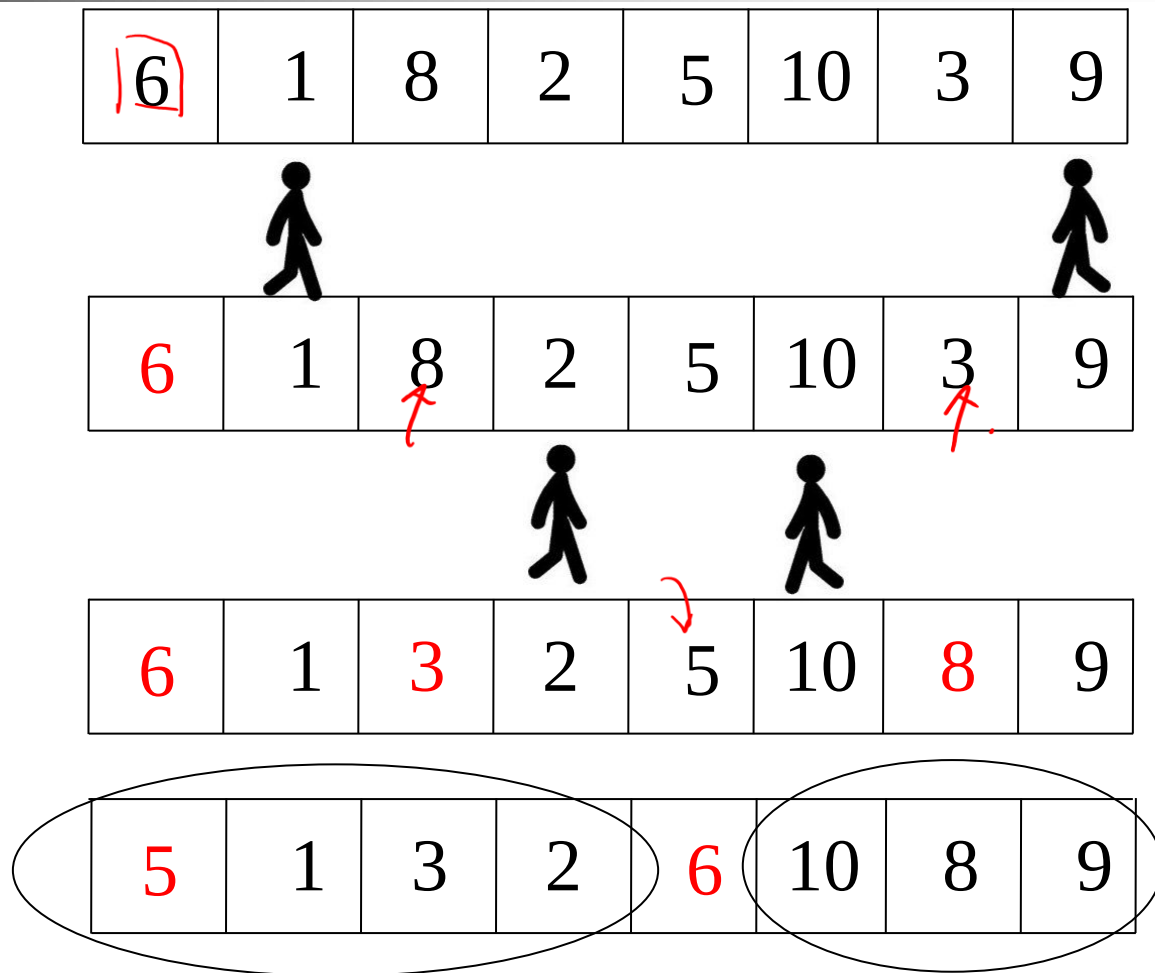
Step2: 其他数字按照和基准数的大小关系分成两部分

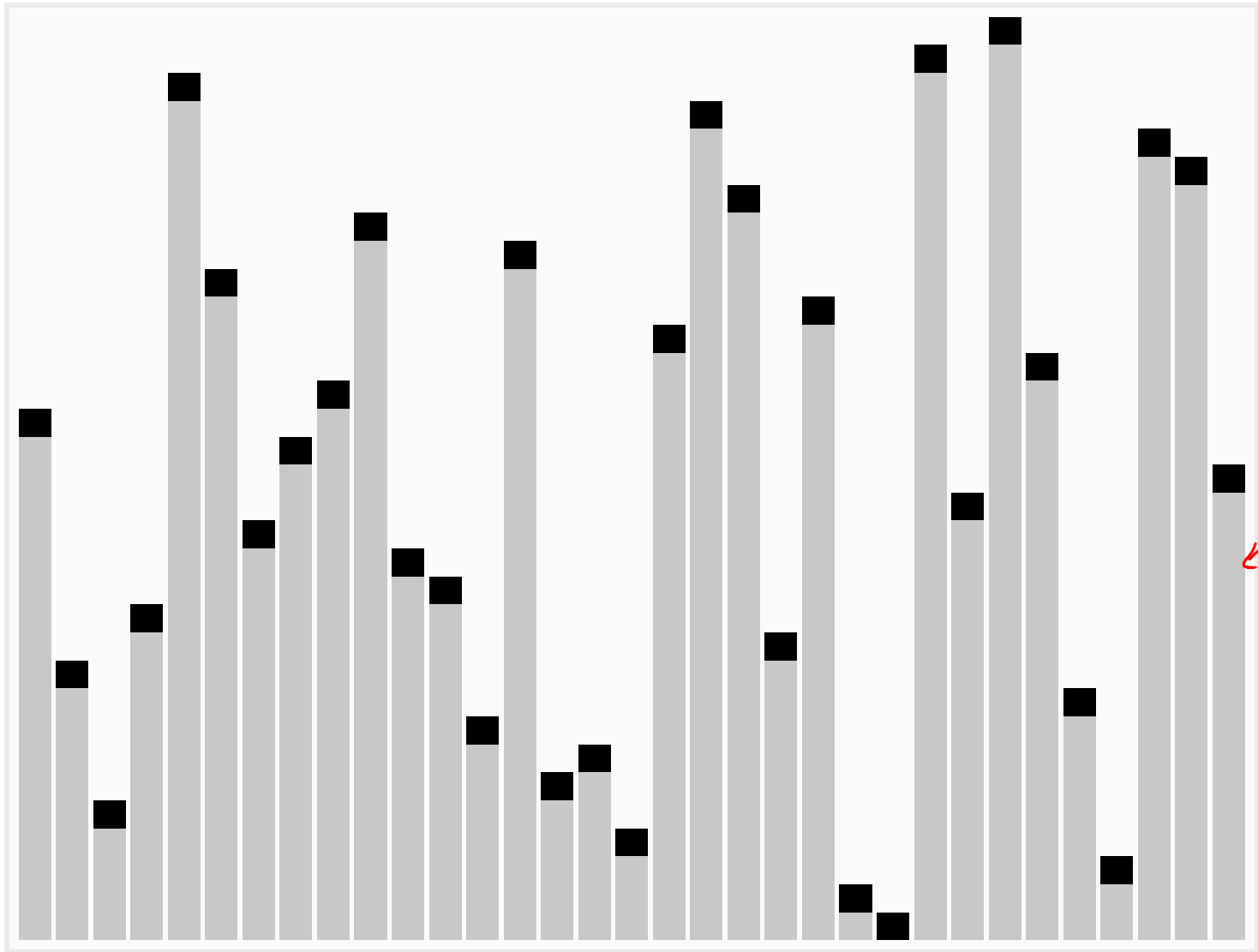


Step3: 分别递归

快速排序：一种双指针的实现方式

基准数6





<https://en.wikipedia.org/wiki/Quicksort>

快速排序

■ 需要多少次比较？

■ 最坏情况： $n \times (n - 1) / 2$

■ 平均情况：？

■ 记 $T(n)$ 代表 n 个数快排平均意义下的比较次数

$$T(n) = \frac{1}{n} \sum_{k=1}^n (T(k-1) + T(n-k) + n-1)$$

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

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

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

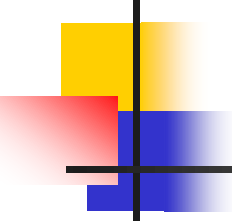
■ $T(n) \approx 2(n+1) \ln n$

$$\frac{1}{n} \times \frac{1}{n-1} \times \frac{1}{n-2} \times \dots$$

$$\frac{1}{n!}$$

n 个数中第 k 大的数
概率： $\frac{1}{n}$

$$(n-1) + (n-2) + (n-3) + \dots$$



小o, 大O记号

- 冒泡排序 $\leq 2 \times \frac{n(n-1)}{2}$ 比较, 交换

- 运行时间 $O(n^2)$

- 快速排序

- 平均运行时间 $O(n \log n)$

- 是 $o(n^2)$

$\approx 2n \ln n$

$$o(g(n)) = f(n) \quad \times$$

小o, 大O记号

■ $f(n) = o(g(n))$: $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$

■ $n^{1.58} = o(n^2)$

■ $n^{1000} = o(2^n)$

■ $(\log n)^{200} = o(n)$

快速: $o(n^2)$

$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq C$

■ $f(n) = O(g(n))$: $\exists$ 常数 $c > 0$, $f(n) \leq cg(n)$ 对充分大的 n 成立

■ $n^{1.58} = O(n^2)$ $10n^{1.58} = O(n^{1.58})$

■ $10^{1000} n = O(n)$

$$f(n) = O(g(n))$$

$\Omega(\cdot), \Theta(\cdot)$ 记号

- $f(n) = \Omega(g(n))$: $\exists$ 常数 $c > 0$, $f(n) \geq cg(n)$ 对充分大的 n 成立

- $n^2 = \Omega(n^{1.58})$ $10n^{1.58} = \Omega(n^{1.58})$

- $f(n) = \Theta(g(n))$:

- $f(n) = O(g(n))$ 并且 $f(n) = \Omega(g(n))$

- $10n^2 - 20n + 45 = \Theta(n^2)$

$20 \frac{n^2}{cgn} + 10n \neq \Theta(n^2)$

$n^2 \log^2 n \neq \Theta(n^2)$
 $\rightarrow \sim \Theta(n^2)$

- 思考: $2^{\Theta(n)}$ 和 $\Theta(2^n)$ 一样吗?

渐进记号小测试

■ 请勾选以下正确的式子

✓ ■ $10n^2 - 100n = O(n^2)$

✓ ■ $n = \underline{O(n^2)}$

✗ ■ $0.01n^3 = O(n^2)$

✗ ■ $n^2 = o(n^2)$

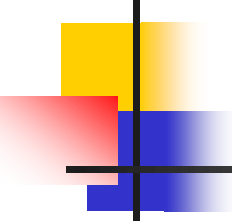
$\frac{n^2}{\log n} = o(n^2)$

$\frac{n^2}{10} \neq O(n^2)$

$\frac{n^2}{10} - 10n \neq O(n^2)$

$O(n^2)$
O
O
O
O
O





排序问题

- 冒泡排序
- 快速排序
- 还有更多的排序算法
 - 选择排序、插入排序、归并排序、堆排序、二叉搜索树、基数排序.....
 - 外部排序

~~5, 7, 10, 12~~
~~7, 10, 14, 25, 36~~

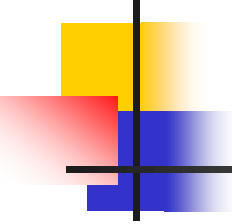
5, 7, 7, 10, 10, 12, 14, 25, 36,

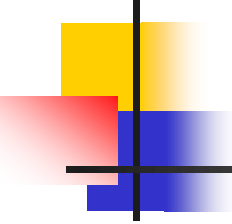
TimSort介绍-设计思想

- 12, 10, 7, 5, 7, 10, 14, 25, 36, 3, 5, 7, 11, 15, 22
- 分成若干单调的部分
- 12, 10, 7, 5, 7, 10, 14, 25, 36, 3, 5, 7, 11, 15, 22
- 合并这些部分
 - 怎么合并两个已经排好序的数串？
 - n 个数的数串和 m 个数的数串合并： $n + m - 1$ 次运算

$m=1$

TimSort介绍-设计思想

- 
- A horizontal line with three colored segments labeled A, B, and C above it. Segment A is green, B is red, and C is blue.
- 12,10,7,5,7,10,14,25,36,3,5,7,11,15,22
 - 合并这些部分
 - 合并顺序会影响效率
 - 先AB合并: $(4+5-1)+(9+6-1)=22$
 - 先BC合并: $(5+6-1)+(4+11-1)=24$
 - TimSort: 给出了一种巧妙的合并顺序
 - 最坏情况 $O(n + n \log \rho)$
 - ρ 是分成的单调的段数



TimSort介绍

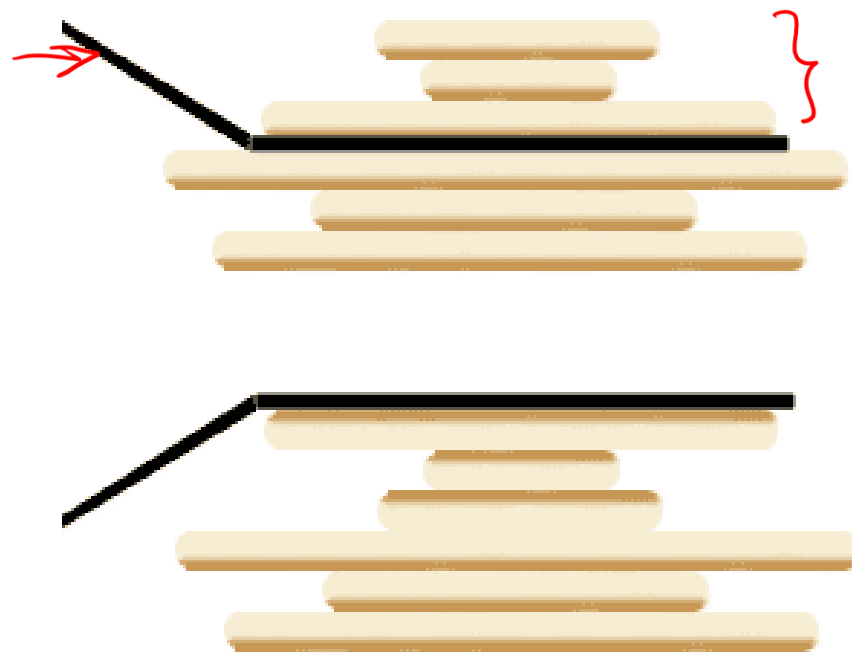
- 快速排序 QuickSort
 - 由Tony Hoare 1959年提出
 - 广泛应用在如C、Java的标准库里面
- TimSort
 - 1993年提出，2002年由Tim Peters实现
 - 应用在Python 2.3, Java SE 7, Android平台.....
 - 最坏情况 $O(n + n \log \rho)$, 2018年证明

思考题



■ 翻煎饼问题(Pancake Sorting)

一个厨师做了一叠大小不同的煎饼，他要不断从上面拿起几个煎饼翻到下面。假设有 n 个煎饼，厨师需要翻动多少次，才能把煎饼按从小到大排好？



翻煎饼问题

■ 2, 1, 4, 5, 3



■ 5, 4, 1, 2, 3



■ 3, 2, 1, 4, 5



■ 1, 2, 3, 4, 5

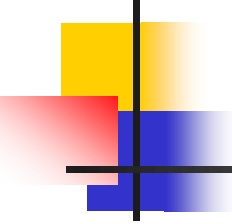


4, 1 | 2, 5, 3

1, 4, 2, 5, 3,

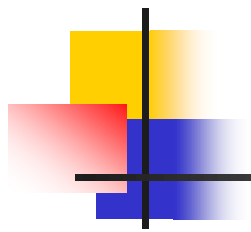


$2n$ - 常数
 $1.9n$



算法思维

- 排序算法
 - 冒泡排序
 - 快速排序
- 小o，大O记号
- 思考题：
 - 翻煎饼问题
 - $2^{\Theta(n)}$ VS $\Theta(2^n)$



谢谢！