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

2018-5-4

思考题

- Mr. S, Mr. P都具有足够的推理能力。约翰教授写了两个整数M和N（3≤M,N≤100)，并把M+N的值告诉了S先生，把M*N的值告诉了P先生。约翰教授问S先生和P先生："你们能从已知的信息确定M和N的值吗？"

  S先生:" 我知道你不知道，我也不知道。"

  P先生:"现在我知道了。"

  S先生:"我也知道了。"

  请问，M和N是哪两个数？ (16,13)

- 记s=M+N, p=M*N.
- 由S可以断言P不知道(M,N)可知：p不可能是
- 1）两个素数（可以相同）的乘积；
- 2）一个素数的3次方；
- 3）4乘以一个素数；
- 4）2乘以一个素数的平方；
- 5）一个>50的素数乘以另一个不小于3的数；

- 因此s不能是上述分解中的两个数之和：
- 1）所有偶数（可以写成两个素数之和）；
- 2）>56的奇数（可以写成53+N）；
- 3）7=4+3, 9=4+5, 11=4+7, 15=4+11, 17=4+13, 21=4+17, 23=4+19, 27=4+23, 33=4+29, 35=4+31, 39=13+26, 41=4+37, 45=4+41, 47=4+43, 51=4+47.
- 可能的和：13,19,25,29,31,37,43,49,53,55
- ……………
- (16,13)

陆润宇的解答：
由于S说他不知道这两个数，于是M+N只可能处于8到200之间。

由于S说P在他说这句话之前不知道这两个数，所以M+N一定满足以下性质：
对于任意满足m+n=M+N的m和n，m*n的分解不唯一。

这样由程序筛选得M+N属于集合A={13,19,25,29,31,37,43,49,53,55}。

接下来，由于P说他现在知道了，于是M*N满足以下性质：
在M*N的所有分解中，仅存在唯一分解m*n=M*N，使得m+n属于A。

那么枚举A中任意元素的任意拆分m+n，对m*n的出现次数进行累加。记那些出现次数为1的m*n构成的集合为B，这样M*N属于集合B（由于|B|较大，在这里不进行列举）。

最后，由于S说他现在也知道了，于是M+N满足以下性质：
在M+N的所有分解中，仅存在唯一分解m+n=M+N，使得m*n属于B。

同样对于B中任意元素的满足m+n属于A的唯一拆分m*n，对m+n的出现次数进行累加，这样出现次数为1的m+n即为M+N，且相应的m和n即为M和N。

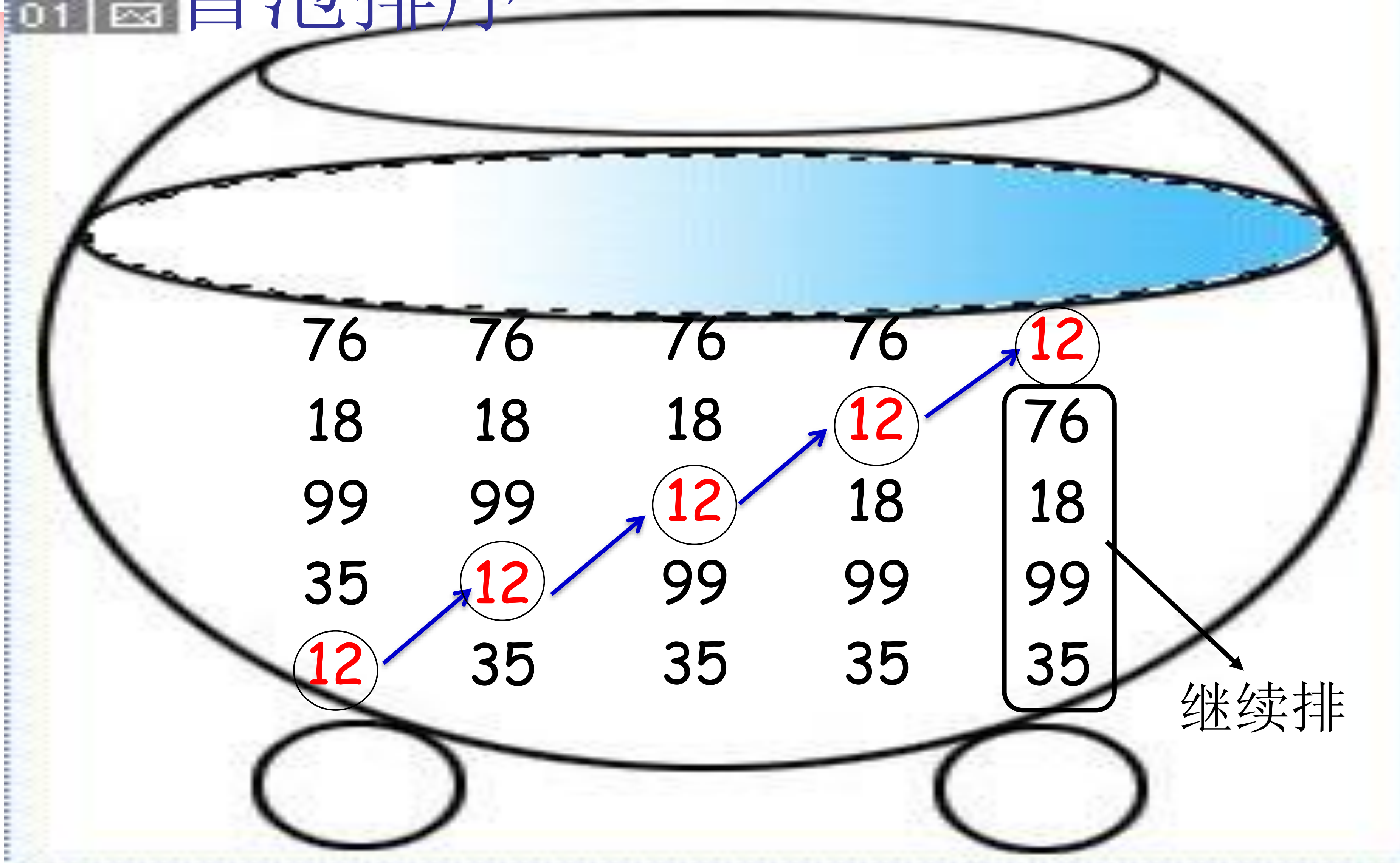
程序运行得唯一解M=13,N=16。

算法思维

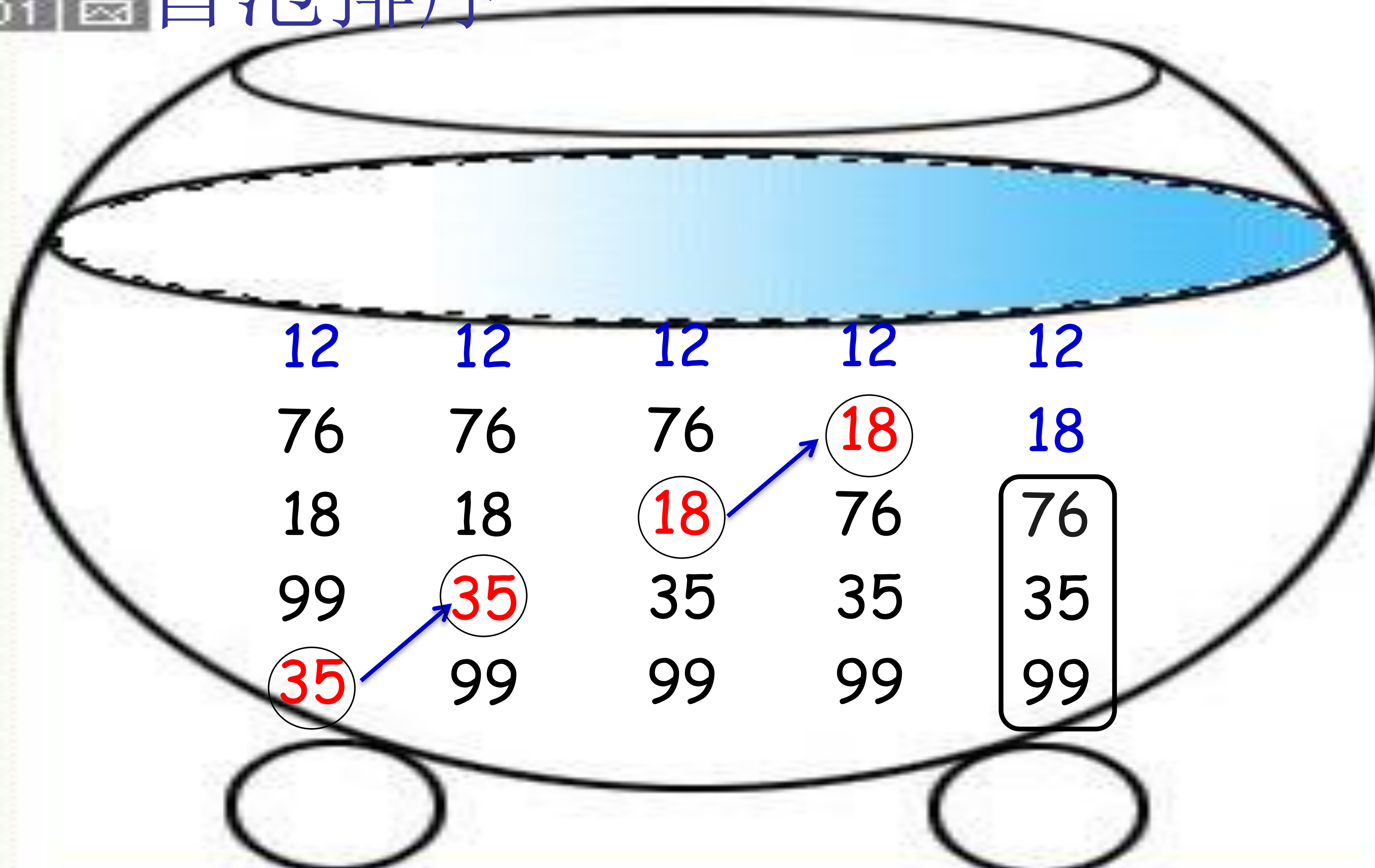
引入：排序

- 任务：给定n个正整数，把他们从小到大排起来
- 请思考在打斗地主、升级、桥牌……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


76 76 76 76 12
18 18 18 12 76
99 99 12 18 18
35 12 99 99 99
12 35 35 35 35
继续排

冒泡排序



12 76 18 99 35
12 76 18 35 99
12 76 18 35 99
12 18 76 35 99
12 18 76 35 99

算法

- 有穷性。
- 确切性。
- 输入。
- 输出。
- 可行性。

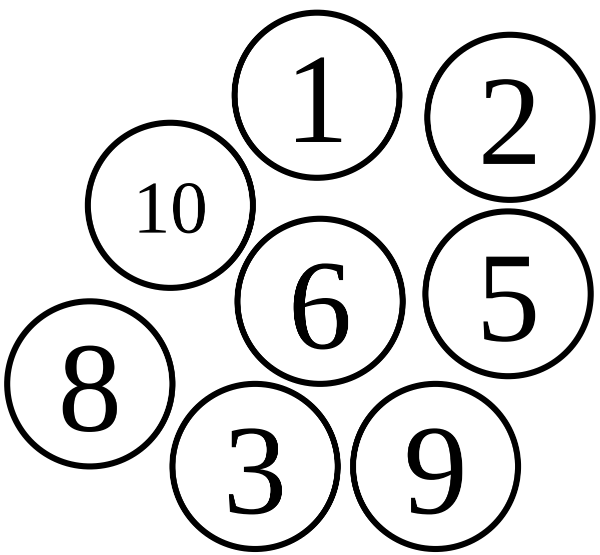
冒泡排序

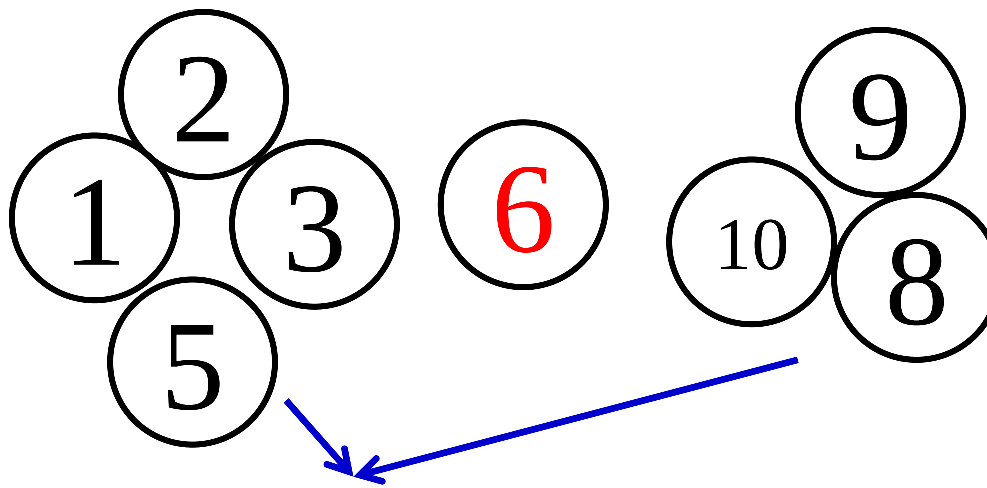
- 需要n-1+n-2+…+1=n*(n-1)/2次比较操作
- 最坏情况下需要n*(n-1)/2次交换操作
  - n, n-1, …, 2, 1
- 能不能更快?

快速排序

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

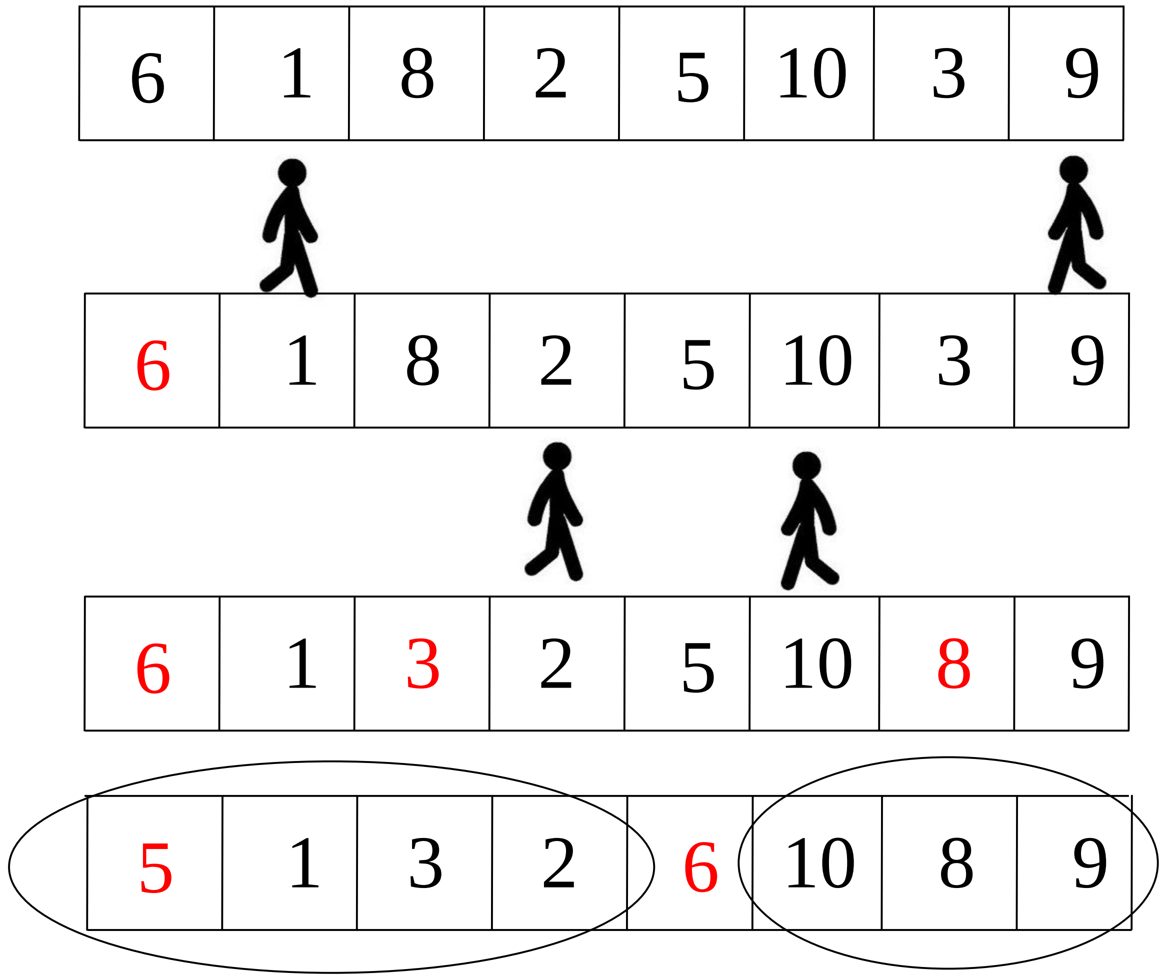


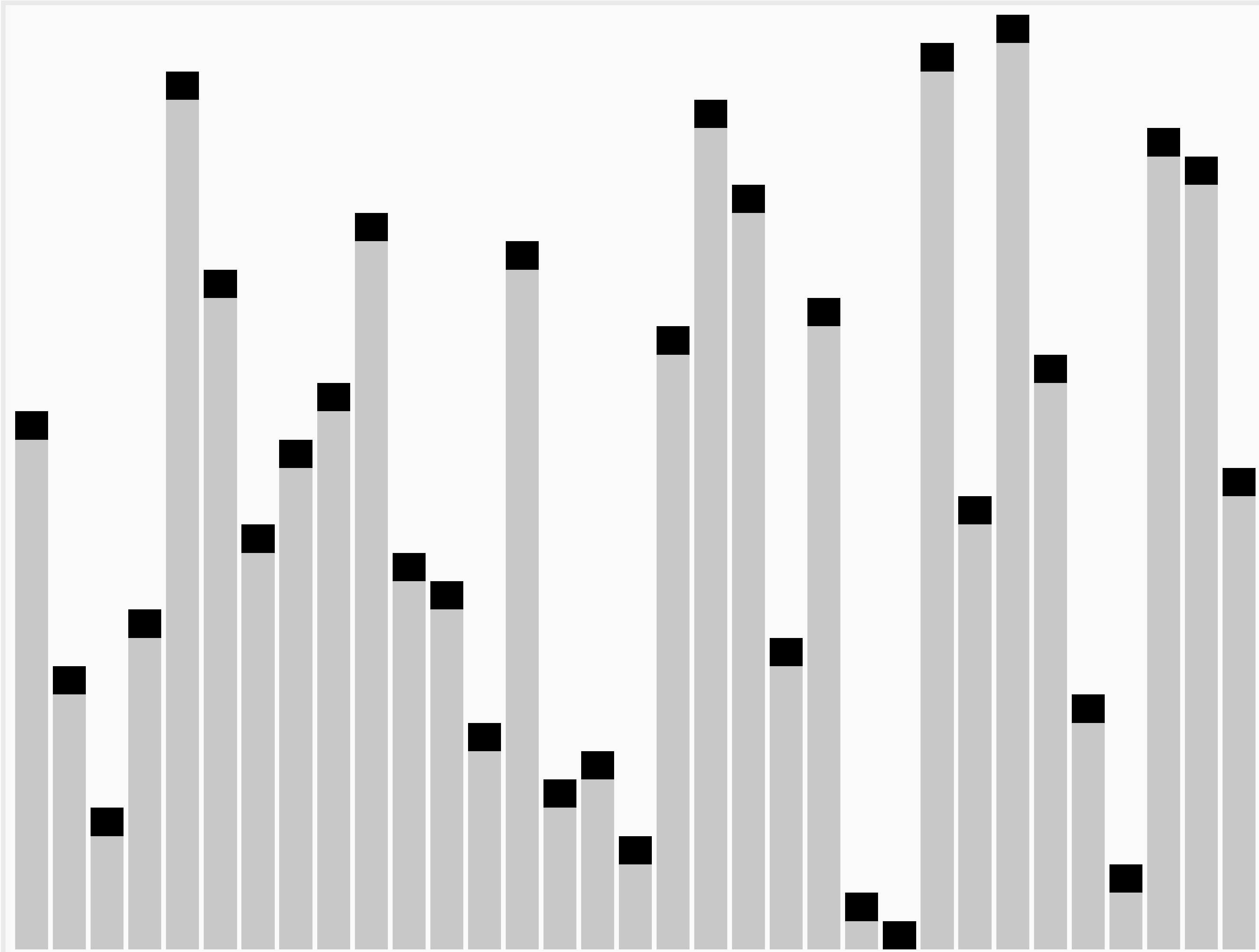


Step3：分别递归

快速排序

基准数6





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

快速排序

- 需要多少次比较？
  - 最好情况：$n\log_2 n$
  - 最坏情况：$n*(n-1)/2$
  - 平均情况：$O(n\log_2 n)$

班级快速排序实验(人体计算机)

- 时间：5月25日上午
- 地点：大礼堂外草地or排球场or大礼堂
- 要求：实现快速排序算法
- 输入：按学号排序
- 输出：按身高排序
- 过程：部分学生充当数据，部分学生充当控制器、监督器等。

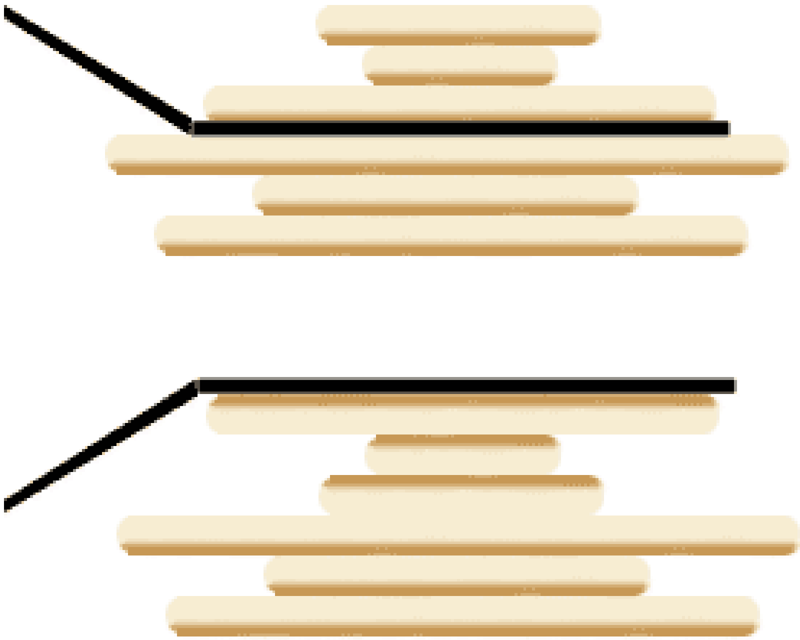
排序问题

- 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

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

- 比尔·盖茨, C.H. Papadimitriou, (1979)

小o，大O记号

- 冒泡排序
  - 运行时间 $O(n^2)$
- 快速排序
  - 平均运行时间 $O(n \log n)$
  - 是$o(n^2)$

小o，大O记号

- $f(n)=o(g(n))$: $\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=0$
  - $n^{1.58}=o(n^2)$
  - $\mathbf{n^{1000}=o(2^n)}$
  - $(\log n)^{200}=o(n)$
- $f(n)=O(g(n))$: $\exists$常数$c>0, f(n)\le cg(n)$ 对充分大的n成立
  - $n^{1.58}=O(n^2)$ $\quad 10n^{1.58}=O(n^{1.58})$
  - $\mathbf{10^{1000}\,n=O(n)}$

$\Omega(\cdot), \Theta(\cdot)$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 $f(n) = \Omega(g(n))$
  - $10n^2 - 20n + 45 = \Theta(n^2)$
- 思考：$2^{\Theta(n)}$ 和 $\Theta(2^n)$ 一样吗？

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！