<!-- Slide number: 1 -->
# 计算机问题求解 – 论题2-2    -  组合与计数
2021年03月08日

<!-- Slide number: 2 -->
Part I
算法分析与计数

<!-- Slide number: 3 -->
问题1：
通常在讨论算法时间代价时我们“数”什么？
“critical operation”:
假设：总代价与critical operations的代价成正比
选择的“合理性”

<!-- Slide number: 4 -->
Selection sort，本质上与“冒泡”一样
顺便问一句: 你能用循环不变量证明此算法正确吗?

![](Picture2.jpg)

![](Picture3.jpg)

![](Picture4.jpg)
critical operation

<!-- Slide number: 5 -->
问题2：
你如何理解这里所体现的“抽象”过程？
我们究竟是在数什么？
If our goal were to solve only Exercise 1.1-1, then our abstraction would have been almost a mindless exercise that complicated what was an “obvious” solution. However, the sum principle will prove to be useful in a variety of problems. Thus, the value of abstraction is that recognizing the abstract elements of a problem often helps us solve subsequent problems.

<!-- Slide number: 6 -->
# 你能解释一下抽象的过程吗？

![](Picture2.jpg)
矩阵相乘
问题3:
多少集合,什么关系?

![](Picture4.jpg)

![](Picture3.jpg)

![](Picture2.jpg)

<!-- Slide number: 7 -->
# 分块计数
The first loop makes n(n + 1)/2−1
Comparisons.
Ask yourself first where the n(n + 1)/2 comes from and then why we subtracted 1

![](Picture2.jpg)

就是排序


![](Picture3.jpg)
数“大间隔”

<!-- Slide number: 8 -->
# 操作计数 与 子集计数
相同的情况，不同的抽象：
  在排序的例子里，对任意含两个元素的子集，我们做一次比较，则比较次数等于n个元素的集合所有的两个元素的子集的个数。

![](Picture2.jpg)

注意: 数操作实际上是数“有序对”，而子集内元素是无序的。

<!-- Slide number: 9 -->
# 多少种密码?

![](Picture2.jpg)
问题4:
几个集合? 相加? 相乘?

<!-- Slide number: 10 -->
问题5:
每个集合大小怎么算?

![](Picture2.jpg)
你能将这个计算推广到一般的原理吗?

<!-- Slide number: 11 -->
# 从数list到数函数

![](Picture2.jpg)
问题6：通俗地说说这是什么意思？
假设list是一步一步生成的，每“步”有多少种选择，全部乘起来。
问题7：这与数函数有什么关系？

### Notes:

<!-- Slide number: 12 -->
# 平面上 n 个点能生成多少三角形

![](Picture2.jpg)

前面排序算法可以通过数子集个数来计数，有什么启发吗？

![](Picture6.jpg)

<!-- Slide number: 13 -->
问题8：
你能解释这个原理的应用吗？

![](Picture2.jpg)
注意这里的三个循环变量。

![](Picture3.jpg)

<!-- Slide number: 14 -->
问题9：
你能否简单叙述一下：为什么在n个数中任取3个不同的数构成的递增序列的集合与所有3个数字构成的子集的集合是等势的？
双射：(i, j, k)  {i, j, k}

<!-- Slide number: 15 -->

![](Picture2.jpg)
三步：
建立一个函数；（注意：不是任意关系都是函数）
证明：这个函数是一对一的
证明：这个函数是满射

<!-- Slide number: 16 -->
Part II
利用等价关系计数

<!-- Slide number: 17 -->
问题10：
你还记得什么是等价关系吗？它和集合分划有什么关系？
给定集合A上的等价关系，所有的等价类构成集合A的一个分划。

<!-- Slide number: 18 -->
# 如何将等价关系与计数相关联
问题：
用2种颜色给5个对象着色，并保证每种颜色最少用于2个对象，有多少种不同的着色法？

问题11：
这个问题其实非常简单，你能直接给出答案吗？
但是如果我们定义一种等价关系，使得等价类的个数就是不同的着色方案的个数，那就可以通过“抽象”推出一个更加有用的解法。
5个对象任意排列方式有5！=120种，如果用颜色代码加标号，有哪些可以被认为是“相同”的着色方案呢？

<!-- Slide number: 19 -->
# 给出一个等价类

![](Picture2.jpg)
(Quotient Principle)
If an equivalence relation on a p-element set S has q classes each of size r, then q = p/r.
这里： p=120,  r=12,  q=120/12=10

<!-- Slide number: 20 -->
# 你能解释下列问题的等价类吗

![](Picture2.jpg)

![](Picture3.jpg)

![](Picture4.jpg)
问题12：
为什么上面问题中等价类大小为n(这里是4), 而下面问题中则为2n?

<!-- Slide number: 21 -->
# Multiset ：允许相同元素的集合
把k册相同的书放置在n层书架上（每层的空间可以放得下所有的书），有多少种不同的放置方式？为什么可以用multiset建模？
How many  k-element multisets can we choose from an n-element set?
问题13:
为什么不能直接采用上述基于等价关系的方法求解？

<!-- Slide number: 22 -->
# 间接求解：借助书架问题

![](Picture2.jpg)
构造出需要的等价类

<!-- Slide number: 23 -->
# 书架上的排列问题
暂且假设每本不同

![](Picture2.jpg)

![](Picture3.jpg)

![](Picture4.jpg)
随着已放置书增加，可选的位置也在“均匀”增加

<!-- Slide number: 24 -->
# 回到Multiset问题
对应到书架问题：我们只关心书在各层书架上的分布，可以当作k本书是一样的，所以任何一种排列等完全等价。

![](Picture2.jpg)

<!-- Slide number: 25 -->
# 课外作业
CS pp.8-: 9, 13
CS pp.20-: 15
CS pp.30-: 6, 9, 14
CS pp.54-: 8, 10, 15