

## 团队

做毕设，摸了。

## 个人

### zzh

[Codeforces Round #645 \(Div. 2\)](#): pro: 5/6/6 rk: 80/18169

[Educational Codeforces Round 88](#): pro: 5/6/6 rk: 159/14430

学习了一下 SA

### pmxm

本周没有打比赛

### jsh

## 本周推荐

### zzh

Python 众所周知 Python 的整数是无限精度的，很多同学也可能知道 Python 有 decimal 处理浮点数。不过可能知道 Fraction 的人就不那么多了。例如 [Educational Codeforces Round 88 C](#) 这样的题，如果用 C++ 写可能会累死累活写 20 分钟，甚至还容易写错。而使用 Python 则可以很容易完成。

### pmxm

KM 如何保持复杂度和点数少的那一边一致的问题。

### jsh

#### 错排问题（错位排列）

错排，是个时常会听到，但做到就有点抓瞎的东西。

原始的全错位排列问题的做法有很多种，但要记得了解到做法的本质，因为出题通常就是某种做法的本质没变，在条件上稍加改动而已。

具体的可参考 [错排问题 - 维基百科](#) 和 [错排公式 - 百度百科](#)

## 牛客练习赛64 - D - 宝石装箱

### 题目链接

题意：

共  $n$  个编号的宝石和  $n$  个编号的箱子  $[1, n]$ 。每个箱子要装有恰好一个宝石，但第  $i$  个宝石不能放在第  $a_i$  个箱子里。

问有多少种装箱的方案数。取模。

$a_i$  不是个排列，可能有重。

错排改了改，每个物品的限制虽然还是一个，但限制可能有重复。

范围上看  $\mathcal{O}(n^2)$  就能过。

那我们考虑一下容斥，记  $A_i$  为每个箱子要装有恰好一个宝石的情况下，第  $i$  个宝石放在了第  $a_i$  个箱子的方案集合，方案数就是  $|\bigcap_{i=1}^n A_i|$

$$|\bigcap_{i=1}^n A_i| = |S| - \sum_{i=1}^n |A_i| + \sum_{1 \leq i < j \leq n} |A_i \cap A_j| - \dots + (-1)^n |A_1 \cap \dots \cap A_n|$$

哦天哪，我们把所有  $S_j$  个  $A_i$  的交集的大小的和，记为  $S_j$  即有  $S_j$  个宝石放错，其余随便放的方案数。问题的答案就是  $\sum_{i=0}^n (-1)^i S_i$

这个  $S_j$  很好算。记第  $i$  个箱子有  $b_i$  个宝石不能放，那么对于放错宝石的数量贡献与否，可以得到多项式  $\prod_i (1 + b_i x)$

记展开的第  $i$  项系数为  $c_i$  即  $i$  个宝石放错的玩法。我们乘上其余宝石乱排的方案数，有  $S_i = c_i (n - i)!$

那个多项式展开就是个背包  $\mathcal{O}(n^2)$  DP 一下，剩下的都好做。

zzh's comment 多嘴一下 FFT 甚至可以做到  $\mathcal{O}(n \log^2 n)$

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:intrepidword:2020.05.22-2020.05.28\\_%E5%91%A8%E6%8A%A5&rev=1590914682](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:intrepidword:2020.05.22-2020.05.28_%E5%91%A8%E6%8A%A5&rev=1590914682)

Last update: 2020/05/31 16:44