

Contest 20181015-16

Curtis Nishikino

摘要

A Summary of NOIP Virtual Contest.

关键字 Strings, Mathematic, Segment Tree, Matrix, Floyd

目录

1	Day 1	3
1.1	Problem A replace	3
1.1.1	Description	3
1.1.2	Solution	3
1.2	Problem B String	3
1.2.1	Description	3
1.2.2	Solution	3
1.3	Problem C Goddess	4
1.3.1	Description	4
1.3.2	Solution	4
1.3.3	Another Solution	5
2	Day 2	6
2.1	Problem D Lgg	6
2.1.1	Description	6
2.1.2	Solution	6
2.2	Problem E Biscuits	6
2.2.1	Description	6
2.2.2	Solution	6

2.3	Problem F Tesseract	6
2.3.1	Description	6
2.3.2	Solution	7
3	Summary And Experience	7

1 Day 1

1.1 Problem A replace

1.1.1 Description

定义一个 k 位二进制数的反转操作, 如 $101100 \rightarrow 001101$, 求 n 个二进制数反转后的结果.

1.1.2 Solution

一开始看到暴力有 90 分, 就码了循环展开, 然后丢掉了, 也没多想.

其实可以预处理 $k/2$ 位的反转结果, 然后再交换高低位的结果, 就是最终的结果.

也可以使用蝴蝶变换.

1.2 Problem B String

1.2.1 Description

对于一个字符串 A , 可以以最后一位为中心, 将之前各位反转, 得到一个新的字符串. 操作可以重复若干次.

现在给出一个字符串 S , 求在 S 的前缀中, 有多少个位置可以通过操作, 使得 S 是结果的前缀.

1.2.2 Solution

考虑每一次操作都会产生一个奇回文串, 就从回文串的性质入手解决问题.

首先一个显然的性质是, 如果一个位置的回文边界是 R , 那么这个位置一定合法. 需要考虑的是不到 R 的情况.

尝试令一个串反转后可以到达一个能继续翻转的位置, 这意味着首先回文边界要落在一个已经被标记合法的点上, 其次回文左边界应为 1, 这样才能满足整体反转后仍然合法.

使用 Manacher 算法, 计算回文半径, 倒序扫描一下串 S , 就可以得到答案.

1.3 Problem C Goddess

1.3.1 Description

计算

$$\sum_{k=1}^n k \cdot \sum_{j=1}^k \sum_{i=0}^{n-1} \binom{i}{k-j} \binom{n-i-1}{j-1} \pmod{10^9 + 7} \quad (1)$$

1.3.2 Solution

首先考虑最里面的 Σ , 两个组合数乘起来再求和, 就相当于将整个线段划分成两部分, 枚举 i 作为左部分的端点, 强制选择 $i+1$, 其余作为右部分, 这样我们相当于是在 $n-1$ 中选择 $k-1$, 再枚举第 k 个位置, 相当于不重不漏地考虑了 $\binom{n}{k}$.

这样再套上第二个 Σ , 就是 $k \times \binom{n}{k}$.

根据组合恒等式, 有:

$$\begin{aligned} k \times \binom{n}{k} &= k \times \frac{n!}{k!(n-k)!} \\ &= \frac{n!}{(k-1)!(n-k)!} \\ &= n \times \frac{(n-1)!}{(k-1)!((n-1)-(k-1))!} \\ &= n \times \binom{n-1}{k-1} \end{aligned} \quad (2)$$

再来看我们得到的式子:

$$\sum_{k=1}^n k \cdot k \cdot \binom{n}{k} \quad (3)$$

可以套用两次组合恒等式:

$$\begin{aligned}
\sum_{k=1}^n k \cdot k \cdot \binom{n}{k} &= n \cdot \sum_{k=1}^n k \cdot \binom{n-1}{k-1} \\
&= n \cdot \sum_{k=1}^n (k-1) \cdot \binom{n-1}{k-1} + \binom{n-1}{k-1} \\
&= n \cdot (n-1) \sum_{k=1}^n \binom{n-2}{k-2} + n \cdot \sum_{k=1}^n \binom{n-1}{k-1} \quad (4) \\
&= n(n-1) \times 2^{n-2} + n \times 2^{n-1} \\
&= n(n-1) \times 2^{n-2} + 2n \times 2^{n-2} \\
&= n(n+1) \times 2^{n-2}
\end{aligned}$$

这用到了另一个性质:

$$\sum \binom{n}{i} = 2^n \quad (5)$$

Q.E.D.

1.3.3 Another Solution

考虑将最后一个 Σ 提前, 得到:

$$\sum_{i=0}^n \sum_{j=1}^{k-1} \binom{i}{k-j} \times \binom{n-i-1}{j-1} \quad (6)$$

这样的意思就是把一个 $n-1$ 的线分成两部分, 在这个固定长度下, 枚举左边选 $k-j$ 个, 右边选 $j-1$ 个, 这样由于最内侧的求和号计算时外部求和号是固定的, 所以组合意义很明确, 为 $\binom{n-1}{k-1}$, 这个式子被计算了 n 次, 也转化成了 $n \times \binom{n-1}{k-1}$.

2 Day 2

2.1 Problem D Lgg

2.1.1 Description

有一个字符串 A , 向后复制得到 B , 在任意一个位置 (可以是首尾) 插入一个字符得到 S , 求可能的原串 A .

2.1.2 Solution

枚举哪一个字符是插入的, hash 数组支持删去后快速合并, 写一个字符串哈希即可解决.

2.2 Problem E Biscuits

2.2.1 Description

有三种物品, 价值为 $A, B, A + B$, 可以放在 n 个格子上, 一个格子可以放一种, 也可以不放, 求价值和为 k 的本质不同方案数.

2.2.2 Solution

事实上第三种物品可以视作在一个格子上同时放置了 A, B 两种物品, 这样我们就可以分开考虑两种物品的贡献了, 使用组合数计算, 可以保证每个格子至多只有一个 A 和 B , 符合题意, 我们只要求出有 i 个 A 时, B 的数量, 在两者均小于 n 时, 随意放置, 并将结果相乘, 即可累加进答案.

关键在于得到 A, B 互不影响的结论.

可以枚举 $i \in [0, n]$, 也可以使用扩展欧几里得算法.

2.3 Problem F Tesseract

2.3.1 Description

有九个点, 其中有一些有向边, 每条边有权值和优先级.

每次询问给出一些可用优先级区间集合 $\{S\}$, 要求在满足优先级递增的情况下, 计算最短路, 或不可达.

$$m \leq 100000, \sum |S| \leq 10000$$

2.3.2 Solution

点数不多，考虑对每个优先级单独构造一个邻接矩阵，由于优先级一定不降，所以最终的路径一定是在每个矩阵走若干步，再进入下一个矩阵，或者跳过一些矩阵，根据弗洛伊德的运作方式，我们可以重定义矩阵的乘法，为走一步由一个矩阵进入下一个矩阵后的任意两点最短路，这样我们发现矩阵是可以顺序合并的。

区间合并和查询需要用线段树维护，在每个节点储存一个矩阵就好了。

一个注意事项是，询问区间可能是乱序的，要将其排序来获得合法顺序。

3 Summary And Experience

这次考试其实不是很理想，和机房最高分差出了 200 多分，分析错因，首先就有将近 100 分的低级错误失分，这是由省二上省一的关键。

C 题不是没有考虑到组合意义，但由于缺乏快速反应和证明能力，导致没有进一步推出算式，明明知道是一道推式子题，却放了近两个小时没有解决，今后要加强这方面的训练。当然，考场上可能不容易证明其复杂组合意义，事实上 AC 此题的选手也没有直接证明题面求和式的组合意义，而是运用一些数学技巧，将其转化成了易理解甚至不需要理解组合意义的式子，包括对组合恒等式的不敏感，也是欠缺的地方。

E 题错因是对复杂度的错估和没有发掘题目性质，导致写出扩欧算法后不知道如何使用，甚至觉得是错误的，导致失分。

F 题首先是一开始的思维方向就错了，对优先级的限制，没有想到用顺次合并矩阵处理，而是选用了分层图，这样其实上限只有 50 分，也限制了思路，而且还写挂了。看点数很少，不是没有考虑过弗洛伊德，但是由于不会处理优先级的顺序，与其失之交臂，这说明对矩阵乘法的性质挖掘还不够深，理解还不够透彻，因为如果发现矩阵可合并，想到用一个维护区间的工具来维护是很容易的，关键就在于如何使用矩阵。

总的来说，在接下来的训练中，要加强对思路的训练和对各种工具的理解，有计划地重新学习各种知识和模板，只有二十多天了，这项工作应该在十天左右完成。