ZOJ Monthly, January 2019

chenjb

2019 年 1 月 19 日

写在前面的话

感谢大家的捧场！不知道下次月赛是在哪一个世纪了，希望大家今天玩得开心！

今天的题目主要来源于浙大去年的内部个人训练题，再次感谢陈松扬、翁才智，陈诗翰等几位学长的帮助。当然我们也要感谢提供这些题目 idea 的同学，在此就不一一列举了。

恰逢考试周前后，弄这比赛真的有点辛苦的，甚至月赛打到一半我还去考了个毛概 (啥都不会要凉透了)，不过看到大家的参赛热情还是很感慨的。

顺便一提，Little Sub 和 Mr.Potato 都是我的队友，希望以后能让大家做到他们出的有 (du) 趣 (liu) 题目。

下面给出参考题解，它可能有点小错，或者有点意识流，也很有可能和你的做法不同，嘤其鸣矣，求其友声，大家可以多多交流。

难度分布

期望：$A < E < B, I < C, F < G < D < H$

实际：$A < E < B, I < G < D, F < C < H$

总结

感觉前面的题和预期基本一样，有爱的签到题 A，不知道广州的同学们有没有熟悉的感觉，考察大家找规律或推式子的能力。E 是另一个签到题，一个数位 dp 例题，但是也能构造出解哦。

B 和 I 属于需要大家去发现一些基本性质就能做的题，我觉得标准做法就挺漂亮的，B 本身也可以用非常暴力的数据结构做法去完成，但是还是推荐大家去思考优雅的解法。

C 可能有点套路？本身灵感来源于一道上古冬令营的题，然后通过数据范围就能感受到找到单调性降一个 N 复杂度就行了，之后就是码码码。

F 可能是我最喜欢的一道题，特别感谢下郑鸿鹄学长，这道“假”博弈题你耐心推完之后会发现它很漂亮。

G 不评价，打印过程有点恶心 (懒癌晚期的我完全不想管)，D 题本身的做法非常偏数学，打表插值又非常套路，本来是个我不喜欢的题目。但是 mstczuo 给我讲了他的做法之后我觉得还行，思路其实挺清晰的，只要对题目的几个性质稍作分析就能大大降低枚举的复杂度了。

H 我只能说 Claris NB！

总体而言，还是希望大家不要只做模板题，也不要出门就闷头码码码，多推推性质，多看看题目的条件会给你带来很大的好处，养成这样的习惯对于提高水平是非常重要的。

A Little Sub and Pascal's Triangle

求杨辉三角第 n 行奇数个数

2018 年广州市高考一模填空题 16 题改编

Solution: 2^p，p 为 n-1 的二进制中 1 的个数，可以用 Lucas 定理来证明。

B Little Sub and his Geometry Problem

做法 1

可持久化线段树。第 p 个版本包含 y = 1 到 y = p 范围内所有点，每个版本维护 x 坐标的区间，求 F(x,y) 时在第 y 个版本中查询区间 [1,x]。维护的信息包括：点的数量 count；　所有点的横纵坐标之和 sum。

易知 F(x,y)=(x + y)*count - sum。

时间复杂度：预处理 O(NlogN) + 询问 O(QNlogN)，有 TLE 风险。

PS: 怕大家说我卡常，但是我觉得这样的做法就不能放它过啊... 虽然因为 zoj 跑得慢，放宽了时限，最后根本就没卡吧，感觉怎么写怎么过。

做法 2

随着移动过程动态维护当前的 count 和 sum。向右走时将横坐标相同、纵坐标当前位置的点加入，向下走时将原先纵坐标相同的点移除。

用了线段树、树状数组神马的都是坏文明！

C Little Sub and his another Geometry Problem

按横坐标排序, 枚举左端点为左边界，利用右侧点作为右边界更新答案，然后更新上下界。

枚举左右线段，更新答案，单次询问 $O(n^2)$

枚举左线段，对于右线段，最大值只能在某两个相邻线段取到。

考虑线段上下端点值的单调性，从右向左枚举左线段时，所有右线段的有效长度不增大，和左线段的距离增大，所以决策点不向右移动，$O(n)$ 了。

D Little Sub and Heltion's Math Problem

出题人写的题解

拓扑是集合上的一种结构。设 T 为非空集 X 的子集族。若 T 满足以下条件：1.X 与空集都属于 T;

2.T 中任意两个成员的交属于 T;

3.T 中任意多个成员的并属于 T;

问题转化为求 n 元集合上拓扑的数量

对 m=2,3,4,5,6 分类讨论

无脑做法

对于每种情况分别打表跑出合适的项数，套入 Berlekemp Messy 快速求解递推式第 k 项即可。

mstczuo 的做法

感谢 mstczuo 友情提供该做法，我觉得很资瓷!

这道题可以看作每个队伍的粉丝是一个集合，一共不超过 6 个集合，求满足题目性质的方案个数。根据题目条件可以证明:

1. 一定有一个集合是所有集合的并
2. 一定有一个集合是所有集合的交
3. 不存在相等的集合
4. 最大的集合一定是全集（否则会存在某个粉丝不喜欢任何队伍）
5. 最小的集合一定是空集（否则会存在某个粉丝喜欢所有队伍）

除了全集和空集，剩下的集合最多有四种，因此粉丝最多有 $2^4 = 16$ 种类型。此时可以暴力枚举这 16 种类型是否存在，然后判定这些情况的合法性，这部分可以预处理。此时对每种情况，问题变成了 n 个物体属于 k(k<16) 种颜色的方案数，组合数学直接算即可。

E Little Sub and Mr.Potato's Math Problem

数位 dp 例题，也有构造的做法。

F Little Sub and a Game

线性基

即倘若将某个 n 位数当作一个 n 维向量（且每维均只能取值 0 或者 1），则由我们的线性代数知识知道，可以将非常多的向量简化成一个线性空间，通过高斯消元法的操作（原版中即将向量们消元成最简的表示集）（这里从原来的加减变成了异或），而在本题中，每个数字高斯消元的运算次数即是位数，即是 log(maxai)，有了线性基，我们就可以在 nlog(maxai) 的时间效率下处理出一个线性空间，不同的向量分别负责不同的二进制位，在这样的形式下，每个数字的选择具有很方便贪心的性质。

然后我们如果要查询能不能取得某个数字，就可以从线性空间的首位开始（因为这样的顺序会在每一步确定一位）查询是否能够组成目标数字，当然相应的我们也可以用它来求最大值和最小值。

转化

若只考虑 Alice 一个人，那么如何转化?

线性基是对每个数字选择或者不选，在本题中却是选择 A 和 B 中的一个，这通过一个简单的转化就可以变成原始的线性基问题：先假装选了 A，再把 A xor B 的数值当作一个向量存到线性基里，那么通过选或者不选 A xor B 就相当于选择 A 还是选择 B。

解法

其实“两个人”只是个吓唬人的幌子，甚至不能算博弈论的题目，因为相当于 Alice 先从自己的线性空间中选择了自己的数字，然后由 Bob 选择他的数字，因为信息透明，Alice 选择的任何一个数字其实都可以方便地得知 Bob 的最优选择，仅此一轮，应该不叫博弈论。

从最高位开始考虑就行了，对于每一位，双方相应的选择可以按以下条件分为 8 种情况：1. 该位为 0 或 1；2. Alice 能否处理该位；3. Bob 能否处理该位

可以知道：（为了方便我们记状态为 (0|1,0|1,0|1) 的形式）

(0,0,0) 和 (1,0,0) 什么都不能做

(0,0,1) 和 (1,1,0) 都是没有必要砸自己的脚的

(1,0,1)Bob 显然会选择 xor

(0,1,0)Alice 显然会选择 xor

(0,1,1)Alice 操作 xor 后 Bob 必定会跟风处理 xor，Alice 也可以选择不 xor，那么 Bob 也就不会动

(1,1,1)Alice 可以主动先 xor 来决定得到的结果，也是 Alice 来决定的，类似处理即可

以上的八种情况只有 (0,1,1) 和 (1,1,1) 是结果没确定的，而这里的结果是 Alice 可以决定的，即相当于将原来的 Alice 管理该位的线性基转变成 xor 了 Bob 对应的

线性基之后的结果，那么在按位处理的时候删除掉 Alice 管理该位的向量，加上 xor 后的就可以了。

G Little Sub and Tree

0.1 Dp 做法

感谢 Claris 提供题解！

显然最优解一定选的都是叶子。假设已经选出了一些叶子，那么将这些叶子两两之间的路径上的点都覆盖住，没被覆盖的每个连通块如果不是链，就不能通过距离区分这些点。所以我们的目标是选择尽量少的叶子，使得被它们两两之间的路径覆盖之外的部分都是链。如果 x 点被覆盖，那么 x 的邻居里最多只能有一个点没被覆盖，且那个点对应的子树必须要是叶子。

任取一个不是叶子的点作为根，特别地当 $n = 2$ 时不存在这种点，可以特判。然后树形 DP，设 $f[i][j]$ 表示考虑 i 的子树，i 点被覆盖住，i 的儿子里有 j 棵子树没被覆盖时，最少需要选择多少个点，显然根据限制条件 $0 \leq j \leq 1$。转移时需要判断子树是不是链，还需要保证非叶子的点至少连了两条边。

时间复杂度 $O(n)$。

0.2 非 Dp 做法

感谢 mstczuo 提供该做法！

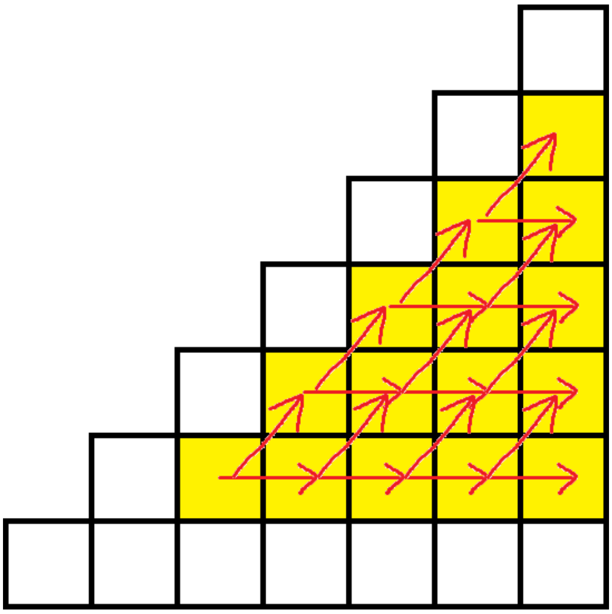
首先将所有的叶子节点选中，然后考虑每个叶子结点，追溯到上面某个度大于 2 的点，如果它没有被标记，则这个叶子可以不用选；否则，必须选。特殊处理一条链即可。

H Little Sub and Zuma

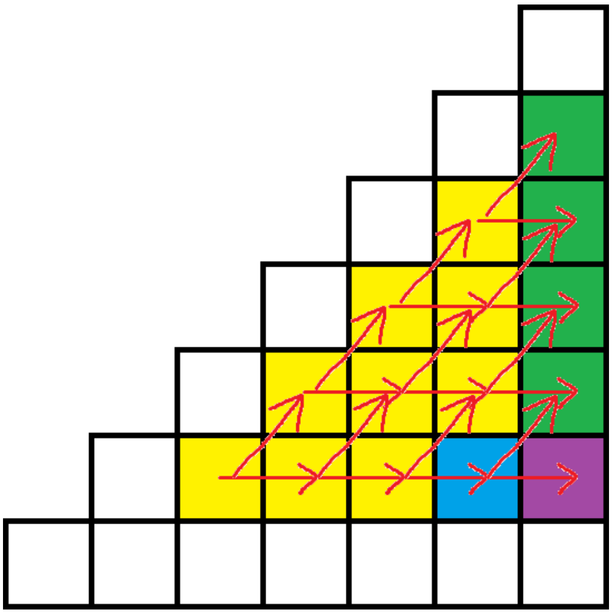
此题最初版本是一个做法 $O(n^6)$ 的题目，而且原题输出小数，迭代足够次数也能够通过。感谢 Claris 提供的做法，我们把它加强到了今天这个样子。

设 $p_{i,j}$ 表示从初始状态转移到串长 i，恶魔在第 j 个位置的概率。考虑转移：$p_{i,j}$ 由 $p_{i-1,j-1}, p_{i-1,j}$ 以及 $p_{>i}$ 转移过来。特别地，当 $(i-1, j-1), (i-1, j)$ 是终止态的时候，它不能进行转移。直接高斯消元复杂度是 $O(m^6)$ 的，不能接受。

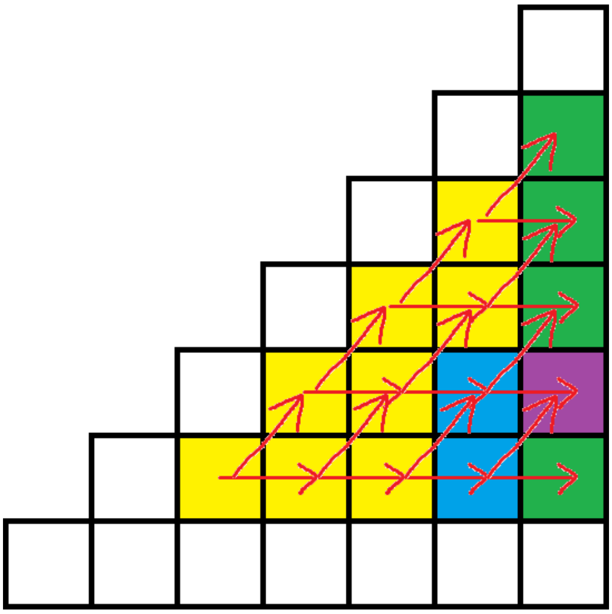
如果仅看 “插入” 的转移，转移情况如下图所示：



假设知道了最右边一列 (绿色的部分)，那么可以直接推出倒数第二列的第二格。因为 $p_{i,2} = p_{i-1,2} \times A + p_{>i}$，所以 $p_{i-1,2} = \frac{p_{i,2} - p_{>i}}{A}$。

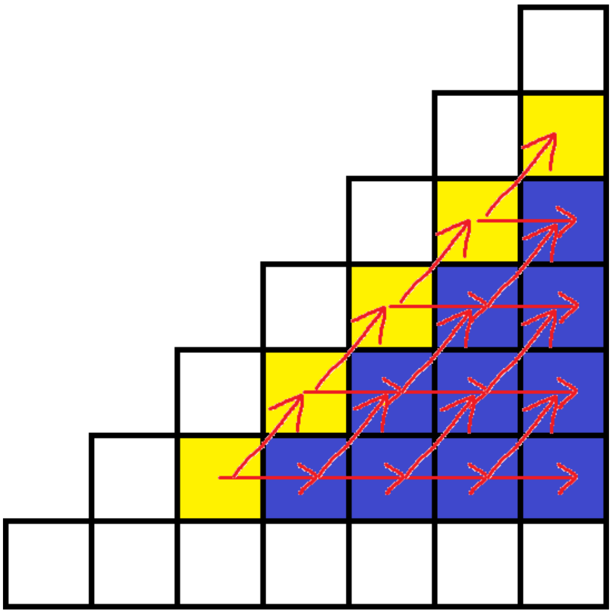


继而可以接着往上推出倒数第二列剩下的部分。因为 $p_{i,j} = p_{i-1,j} \times A + p_{i-1,j-1} \times B + p_{>i}$，所以 $p_{i-1,j} = \frac{p_{i,j} - p_{i-1,j-1} \times B - p_{>i}}{A}$。



如此我们可以将每个不是终止态的格子的概率都用最后一列那 $m-2$ 个量线性表示出来，利用“插入”的转移完成倒推。而“切割”的转移可以通过前缀和加速。至此可以在 $O(m^3)$ 时间内得到每个格子的表达式。

每次倒推消耗掉的是蓝色部分的转移方程：



还剩 $m-2$ 个方程未使用，而我们的未知量恰好有 $m-2$ 个，利用这 $m-2$ 个方程消元求出最后一列的概率即可。求出最后一列后，就知道了所有的概率。

总时间复杂度为 $O(m^3)$。

I Little Sub and Isomorphism Sequences

首先，最长同构子串对总是会重叠的，然后两个子串不重叠的部分必须是同构的。然后不重叠的部分长度为 1，因为由不重叠长度大于 1 的同构子串对总是可以找出不比他短的同构子串对。

所以题目转化成维护相同的数对之间的距离的最大值。对所有的数离散化一下，用 20w 个 set 维护每个不一样的数的位置。，用 1 个 multiset 维护同一个数最右到最左的距离，其中最大值就是答案。