第一部分作业

绍兴市第一中学 张恒捷

2015 年 1 月 12 日

Contents

Chapter 1

codeforces

1.1 263E Rhombus

题目大意

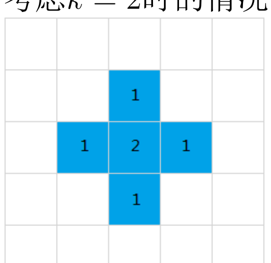
给出一个$n*m$的矩阵，定义函数f
$f(x,y)=\sum_{i=1}^{n}\sum_{j=1}^{m}a_{i,j}\cdot max(0,k-|i-x|-|j-y|)$

给出n,m,k,你需要求在$k \le x \le n-k+1$ 和$k \le y \le m-k+1$ 时f能取到的最大值。

算法

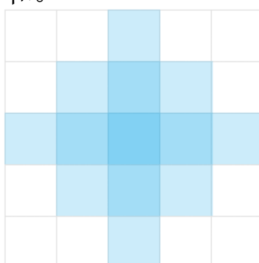
此题普通的方法都显得很麻烦，这里介绍一种非常方便的做法。

考虑$k=2$时的情况

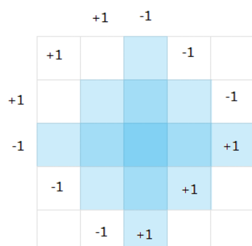


f函数的值相当于每个格子的值乘上它对应的权值。

由于权值的特殊性，f函数的值还可以表示成若干个子矩形内格子权值的和。



到这里就已经有些明显了，再对这些矩形进行差分。就可以表示成若干个前缀矩形的加减。



这里$+1, -1$表示前缀矩形的加减，发现它们都形成了一条斜线。所以此题只要计算4条斜线的值就可以算出每个f的值，进而求出最终的答案。

时间复杂度：$O(NM)$，空间复杂度：$O(NM)$

1.2 293B Distinct Paths

题目大意

有一个$n \times m(n, m \leq 1000)$的方格，你可以染$1-k(k \leq 10)$种颜色。上面已经有一些格子有颜色了，求染色方案数保证：任何从$1, 1$只能向右向下走，走到n, m的路径上任何颜色只能出现一次。

算法

显然$n+m-1>k$是无解的。但是有这个条件去暴搜还是会T的。所以需要使用最小表示法，然后对于一个方案去算可以换的颜色，乘个变换方案加到最终方案数中。

时间复杂度：$O(?)$，空间复杂度：$O(NM)$

1.3 235E Number Challenge

题目大意

设$d(x)$为x的的因子个数。给出a,b,c，求$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(i\times j\times k)$
$a,b,c\le 2000$

算法

用$f[p][i][j][k]$表示最近用的质数是p，a被除到了i，b被除到了j，c被除到了k的方案。后三维实际上只会是a,b,c的约数。大于$\sqrt{max(a,b,c)}$ 的质数在a,b,c中最多出现一次,这些质数可以特殊处理。第一维可以滚存省内存。

时间复杂度：$O(\sqrt{a\times b\times c}log(\sqrt{a\times b\times c})\times 14)$ 其中14 为$\sqrt{2000}$，空间复杂度：$O(\sqrt{a\times b\times c})$

1.4 325E The Red Button

试题大意

N个点，从0到N-1编号，i向$2i\ modulo\ n$与$(2i+1)\ modulo\ n$连边。求从0开始遍历所有数字有且仅有一遍回到0 的方案。

$2 \leq n \leq 10^5$

算法

为了方便，接下来所有编号都是在模N下的，比如x实际表示的是x modulo N。

N为奇数时肯定无解，因为要连向0的点为0与$\lfloor \frac{n}{2} \rfloor$,连向n-1的为$\lfloor \frac{n}{2} \rfloor$ 与n-1，由于自己到自己本身就两遍了，所以$\lfloor \frac{n}{2} \rfloor$要同时连向0与n-1，所以无解。

当N为偶数时，x连出去点是2x与2x+1,$x+\frac{n}{2}$连出去的点是$(x+\frac{n}{2})\times 2 = 2x$ 与$(x+\frac{n}{2})\times 2+1 = 2x+1$，发现x与$x+\frac{n}{2}$ 连出去的点是相同的。

就是说，如果你选择了从x走向2x，那么$x+\frac{n}{2}$只能走向2x+1，反之亦然。

现在，随便选一个满足上述条件的方案，可能会形成不止一个环，现在考虑怎么把两个环并起来。

一个奇怪的定理：如果有两个以上的环，那么一定存在一个x使得x与$x+\frac{n}{2}$在不同的环中。

证明：因为x与$x+\frac{n}{2}$在同一个环中，那么2x与2x+1也一定在这个环里。用这种方法可以推得所有点都在同一个环中。

现在，假设x与$x+\frac{n}{2}$不在同一个环中，x连向2x+A,$x+\frac{n}{2}$连向2x+B（$A+B=1$）。如果把它们反一下，x连向2x+B,$x+\frac{n}{2}$连向2x+A，那么就会把这两个环并在一起。

事实上还有一个更方便的方法：

考虑把x 与$x+\frac{n}{2}$ 看成一个点，把相同的边合并，原图就变成了一张$\frac{n}{2}$个点，每个点两条出边两条入边，题目转换成了经过每条边仅一次的方案，求一遍欧拉路就行了。

时间复杂度：$O(n)$**，空间复杂度：**$O(n)$

1.5 260E Dividing Kingdom

题目大意

一个笛卡尔平面上有n个点，你需要横着划2条线，竖着划2条线，割成9个块，使得9个块中点的个数与给定的个数匹配（不一定要有顺序）。

算法

暴力枚举给定的点数在那个块中，共有9!种情况。对于一种情况，可以推出横竖2条线的位置，然后只要用函数式线段树求得点数与之判断即可。

时间复杂度：$O(nlogn + 9! \times 9 \times logn)$**，空间复杂度：**$O(nlogn)$

1.6 317E Princess and Her Shadow

题目大意

一个方格地图中有m个格子有障碍。A要捉住B。A每次只能往4联通的空的一格内走去，B会和A走相同的方向，不过如果B前面是障碍他就不会走。求A的一种走法使得最终A与B在同一个位置。

设S为所有坐标绝对值的最大值。$m \leq 400$，$S \leq 100$。

算法

如果A与B可以同时走到无限远的地方，那么只要找最外面的障碍，把B卡住就行了。

如果A不存在一条到B的路径，那么肯定无解。

否则A不停的向B走去，最终A一定会和B站在一起的。

时间复杂度：$O(S^2)$，空间复杂度：$O(S^2)$

1.7 325D Reclamation

试题大意

有一个r*c的地图，把左边界和右边界粘起来使得形成一个圆柱，现在要不断地挖去其中的格子，要求任何时候都存在一条从最上方到最下方的路径(四联通)，如果某次操作不满足要求则不做，问最后有多少次操作是成功的。

n为操作次数。

$r, c \leq 3000$, $n \leq 300000$

算法

存不存在从最上方到最下方的路径了，与存不存在一条由障碍构成的八联通的环形路径是等价的。问题就转化为判断是否存在由障碍构成的八联通的环形路径。这个问题可以用并查集轻松解决。

时间复杂度：$O(n\alpha_n)$，空间复杂度：$O(r \times c + n)$

1.8 360D Levko and Sets

试题大意

有n个数ai ，m个数bi。对于一个ai，最初集合中只有1这个数字，然后每次从集合中挑出一个数字x，如果$x \times ai^{bj} mod\, p$ 不在集合中，则加入。最后求这n个集合的并有多少个元素。

p是质数。

$n \leq 10000$，$m \leq 1000000$

算法

对于ai来说，任何一个集合中的元素都可以表示成：$ai^{\sum bj*cj} mod\ p$，设$t = gcd(b1, b2, ,, bj, p-1)$，上式可以写成$ai^{kt}\ mod\ p$，设g是p的一个原根，上式又等于$g^{ri*k*t}\ mod\ p$，另$qi = gcd(ri*t, p-1)$，$g^{k*qi}\ mod\ p$。问题就成了$k*qi\ mod\ (p-1)$有多少。容斥一下即可。这时我们并不需要求出g再求qi，发现i这个集合中有$(p-1)/qi$个元素。我们只要找到最小的d使得$(ai^t)^d\ mod\ p = 1$，就说明i集合元素为d个。所以$qi = (p-1)/d$。d只要枚举p-1的约数即可。

时间复杂度：$O(约数个数^2 + nlogn)$，空间复杂度：$O(n + 约数个数)$

1.9 339E Three Swaps

试题大意

一个长为n的序列a，$a[i] = i$。每次操作选择可以把l-r一段翻转一下。已知最多翻转3次。给出最终序列的样子，求一种合法方案。

算法

如果把连续的一些数字并成一块，翻转一次最多使块数+2，因此最终块数不会超过7。所以可以求出最终序列中的所有块，每次翻转都是一整个块，然后直接暴搜方案。注意最终序列中的一个块的最左端和最右端的数字可能是某次操作后并起来的，所以每个块的最左端与最右端要单独成块。

时间复杂度：$O(n+7^7)$，空间复杂度：$O(n)$

1.10 332D Theft of Blueprints

试题大意

一张n个点的带权图，给出常数k，图满足对于任意一个k个点的子集，仅存在一个点使得它与这k个点都有边，选择这个子集的代价为这些边的边权之和。

现在随机选择一个k个点的子集，求期望代价。

算法

可以通过验证发现k只能成为1,n-1,2中的其中一个。k=1时，图是由若干个大小为2的联通块组成的。k=n-1时，图是完全图。k=2时，图是若干个三角形，这些三角形共用一个顶点。分别判断即可。

也有一种与k无关的方法：枚举重要点，若它的度数$\geq k$，则贡献相应的答案。

时间复杂度：$O(n^2)$，空间复杂度：$O(n^2)$

1.11 286E Ladies'Shop

试题大意

给出n个重量ai。你需要选定k个物品的重量，使得它们在不超过总重量m的情况下做无限背包后能到达的重量与给出的ai 完全一致。若有多种方案则选k最小的方案。

$n, m \leq 1000000$

算法

先判定ai是否合法，只要对于$\forall i,j(a_i+a_j \le m)$，存在k使得$a_k = a_i+a_j$。这一步可以用FFT来快速判断。

至于选的物品，只要在ai中选出那些不会被其他物品组合起来代替掉的就行了。这一步可以用刚才的FFT算出组合方案来判断。

时间复杂度：$O(mlogm)$，空间复杂度：$O(m)$

1.12 335D Rectangles and Square

试题大意

给出n个不重叠的矩形，求一个矩阵的子集使得它们拼接成一个正方形。

$n \le 10^5$，坐标范围≤ 3000

算法

枚举一个矩形让它成为正方形的左上角，然后枚举边长，用预处理的一些东西判断此边长的正方形是否可行。

时间复杂度：$O(3000n)$，空间复杂度：$O(n+3000^2)$

1.13 261D Maxim and Increasing Subsequence

试题大意

将一个n个元素的序列自我复制t遍后得到序列b，求其最长上升子序列。

$n \times maxb_i \le 10^7, n, maxb_i \le 10^5, t \le 10^9$

算法1

DP，$f[i][j]$表示当前最长上升子序列长度为i，末尾以数字j结尾，其末尾最近的位置。可以用前缀和转移。

第一维大小为n，第二维大小为$maxb_i$，转移$O(1)$

时间复杂度：$O(n \times maxb_i)$，空间复杂度：$O(n \times maxb_i)$

算法2

使用经典的二分算最长上升子序列的思想，用f_i表示长度为i的上升子序列其末尾最小的数字。由于题目的特殊性，可以直接维护一张表G，G_i表示数字i加入后会二分到哪个地方。G的维护是$O(ans \times maxb_i)$的。

时间复杂度：$O(n \times maxb_i)$，空间复杂度：$O(n)$

1.14 235C Cyclical Quest

试题大意

给出串S，有n个询问，每次询问给出一个串X，问S中有多少子串与X循环同构。循环同构指的是头尾拼接后的循环串相同。

$|S|, \sum |X| \leq 10^6, n \leq 10^5$

算法

X与Y循环同构可以表示成：Y在X+X(字符串相加)中出现过。因此对串S建后缀自动机，每个询问让X+X在后缀自动机中走。由于只要匹配长度$|X|$就算匹配成功，所以如果当前匹配长度$\geq |X|$时可以往回跳来匹配更多的点。把走到的点去重后答案相加即可。

时间复杂度：$O(|S| + \sum |X|)$**，空间复杂度：**$O(|S| + |X|)$

1.15 311E Biologist

试题大意

n只狗，每只狗有0/1两种性别。改变一只狗有费用。有m个人，一些人是你的朋友。他们会要求一些狗全部变成某个性别。若达到了他们的要求，他们会给你一定的报酬，否则如果那个人是朋友，则你需要花费一些代价。求最大利益。

每个人要求的狗的个数≤ 10

$n \leq 10^4, m \leq 2000$

算法

最小割模型，一只狗变成0连源，变成1连汇。一个人根据他要求的性别，相应地连源或者连汇，然后对应的狗都和他连边。

时间复杂度：$O(Maxflow(|V| = O(n+m), |E| = O(n+10m)))$**，空间复杂度：**$O(n + m)$

1.16 306D Polygon

试题大意

Polycarpus喜欢凸多边形，尤其喜欢每个角角度都相同，每条边长度都不相同的凸多边形。你需要在知晓给定的顶点数的情况下，为他描绘出这样的任意一个凸多边形。

多边形边数$n \leq 100$

算法

这题的算法有很多种。

首先$n \le 4$时无解。

可以使用很多边长接近的边，摆放到只剩一条边的时候，让这条边和第一条边交在一起就可以了。

时间复杂度：$O(n)$，空间复杂度：$O(1)$

1.17 249E Endless Matrix

试题大意

一个无限数阵:

1	2	5	10	17	26
4	3	6	11	18	27
9	8	7	12	19	28
16	15	14	13	20	29
25	24	23	22	21	30
36	35	34	33	32	31

T组询问，每次求内部一个子矩阵的数值和的后10位，而且大于10位的答案需要表示出来。

$T \le 10000$，坐标$\le 10^9$

算法

算答案的话只要将矩阵差分后就很容易了。

麻烦的是判断是否大于10位。

有两种做法：

1.用double判断，速度较慢

2.先去mod很大的数，看看是否大于$10^1$0。速度快但易错，只要多试几次就能保证正确率了。

时间复杂度：$O(T)$，空间复杂度：$O(1)$

1.18 295D Greg and Caves

试题大意

一个由n*m个像素点组成的矩形，每个像素点可以是黑色或白色。

如果符合以下条件，那么矩阵合法：

一.存在一个区间[L,R]($1 <= L <= R <= N$)，使得上至行L、下至行R，每一行都恰有2个黑点，而其他列只有白点。

二.存在一个行号t∈[L,R]，使得：

定义：中间白点是指同一行处在2个黑点之间的白点，$blackleft[i]$表示第i行靠左边的黑点所在列，$blackright[i]$表示靠右边的黑点所在列

1.对于任意$L <= i <= j <= t$，第i行的任意列的中间白点，第j行的同一列都是中间白点。且$blackleft[i] >= blackleft[j], blackright[i] <= blackright[j]$

2.对于任意$t <= i <= j <= R$，第j行的任意列的中间白点，第i行的同一列都是中间白点。且$blackleft[i] <= blackleft[j], blackright[i] >= blackright[j]$

求合法矩阵方案数。

$n, m \leq 2000$

算法

$f[i][j]$表示用了i行，第i行最长且它的长为j的方案数。转移用前缀和。

算方案的时候枚举t在哪里，然后以t为界的两半用f组合在一起即可。

时间复杂度：$O(nm)$**，空间复杂度：**$O(nm)$

1.19 264E Roadside Trees

试题大意

有n个坑，m个操作。

1. 在x位置上建y(≤ 10)高度的树

2. 砍掉前面数过来的第k(≤ 10)棵树。

每个操作后每棵树都会长高1，被砍过的树的坑不会再长树。

每棵树高度两两不同。

每个操作后输出树高度的最长上升子序列长度。

$n \leq 10^5, m \leq 2 * 10^5$

算法

由于每次操作只会影响最多10棵树，因此要维护最长上升子序列关于位置的线段树与关于高度的线段树。每次操作将影响的树全部拎出来重做。如果是操作1，则使用位置线段树求答案，否则使用高度线段树求答案，顺便维护一下另一棵树的信息即可。

时间复杂度：$O(10 * (n + m)logn)$**，空间复杂度：**$O(n + m)$

1.20 286D Tourists

试题大意

共有n面墙，在qi时刻，有一面(0,si)到(0,ti)的墙出现。

有m组询问，每一组问你，有两个人在Ti时刻分别从(-1,0)和(1,0)向y轴正方向向前走，有几秒种他们无法互相看见。

墙会重复或者交叉。

$n, m \leq 10^5$

算法

先用离散+并查集搞出每一段墙最早出现时刻。然后对于l到r，t时刻出现的墙,在出发时间为$-inf->t-r, t-r\ t-l, t-l\ +inf$贡献的答案分别为$0, i-(t-r), (r-l)$，弄个前缀和就好了。

时间复杂度：$O(nlogn+m)$，空间复杂度：$O(n)$

1.21 258D Little Elephant and Broken Sorting

试题大意

一个n个元素的排列，m个操作（x,y）表示50%概率交换第x,y元素。求逆序对期望。

$n, m \leq 1000$

算法

用F[i][j]表示第i个位置上的数字比第j个位置上的数字大的概率。

每个操作过后相应地维护一下f数组就可以了。

时间复杂度：$O(nm)$，空间复杂度：$O(nm)$

1.22 273D Dima and Figure

试题大意

Dima喜欢在一块长方形纸片上作自己喜欢的画。

一块大小为N*M的长方形纸片，包含着N行与M列，并且初始所有格子都是白色的。

Dima每一次作画，会把纸片上一些格子涂黑，并定义这幅“画”就是所涂黑的格子。

Dima喜欢一幅画仅当：

*包含至少一个涂黑的格子。

*所有的涂黑的格子形成一个连通块，换句话说，你可以从任意一个涂黑的格子移动到另一个任意涂黑的格子（一个格子可以移动到四联通的格子里)。

*从一个涂黑的格子(x1,y1)到另一个涂黑的格子(x2,y2)所需的最少移动步数等于$|x1-x2|+|y1-y2|$。

请帮助困惑的Dima，在N*M的长方形纸片上，他能画出多少幅自己喜欢的画呢？你只须回答答案对10^9+7取模就行了。

$n, m \leq 150$

算法

由于上述限制，可推得每一行的黑点都是连续的，且每一行的左右边界都是先单调往外，再单调往内的。

用$f[i][j][k][0..1][0..1]$表示做到第i行，第i行的左右端点在j,k,再用0,1记录左右边界是往外还是往内。转移用前缀和可以优化到$O(1)$。

时间复杂度：$O(n*m^2)$，空间复杂度：$O(n*m^2)$

1.23 283E Cow Tennis Tournament

试题大意

n值奶牛，每只奶牛有一个能力值Si。任意两只奶牛交手总是能力值大的赢。有m个操作，每次操作给出两个整数ai,bi,所有能力值在[ai,bi] 之间

的奶牛，它们互相之间交手的结果会恰好相反。

计算有多少(p,q,r)使得p打败q，q打败r，r打败p。

算法

如果a能打败b，则a向b连边。现在来算不满足题目条件的三元组，答案可以从总方案数减去它得到。

如果一个三元组不满足题目条件，那肯定有一只奶牛有两条入度边，且唯一对应。设ai为能打败i的牛的数量，那么最终答案就等于$n*(n-1)*(n-2)/6-\sum \frac{ai*(ai-1)}{2}$

现在考虑操作的影响，把交手结果写成n*n的01表，一个操作相当于翻转一个子矩阵内的交手结果。这个可以用扫描线+线段树轻松解决。

时间复杂度：$O((n+m)logn)$，空间复杂度：$O(n+m)$

1.24 305D Olya and Graph

试题大意

Olya现在有一张n个点、m条边的有向无权图。现在我们以某种方式给所有点从1到n标号，从而保证原图中任意从u到v 的有向边满足不等式：$u<v$。

现在Olya想知道有多少种方案添加任意数量（可能是0）的有向边，使得该图满足下列条件：

1、从点i出发，可以到达点i+1，i+2，..,n。

2、任意从u到v的有向边满足不等式：$u<v$。

3、两点之间最多有一条边。

4、对于一对点i、j($i<j$)，若$j-i\leq k$，那么从i到j的最短距离等于$j-i$条边。

5、对于一对点i、j($i < j$)，若$j - i > k$，那么从i到j的最短距离等于$j - i$或$j - i - k$条边。

我们认为两种添加边的方案不同，当且仅当存在至少一对点i、j($i < j$)，第一张图中有一条从i连向j的边，而第二张图中没有。

帮助Olya。由于要求的答案可能太大，将答案对1000000007取模后输出。

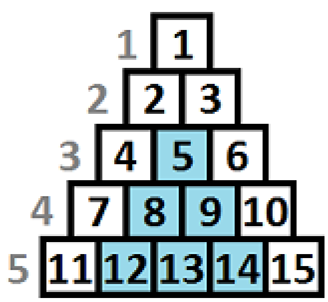
算法

很显然，最终的图就是一条从1到n的链上加一些i到i+k+1的边。而且不能同时存在两条边$(i < j) i \to i + k + 1$与$j \to j + k + 1$满足$i + k + 1 \le j$。先判断原来的有向边是否合法后直接算方案。

时间复杂度：$O(n + m)$，空间复杂度：$O(n)$

1.25 354D Transferring Pyramid

试题大意



一个长成这样的金字塔，你有2种操作：

1.将(i,j)格涂色，费用为3

2.将(i,j)格以下的（如上图蓝色部分）全部涂色，费用为2+格子数

求使给定的k个数全部涂色的最小费用。

$n, k \le 10^5$

算法

将三角形分成斜右上的n条，$f[i][j]$表示前i条中所有点都已经涂完了，且之前第2个操作仍然覆盖了最下面的j行的最小费用。转移时要么加入一个更高的2操作，然后把剩下的较高的点用1操作搞掉。

可以发现2操作不会覆盖在高度大于$\sqrt{6k}$的地方，否则全用1操作都比它优。所以上式j只会到$\sqrt{6k}$。

时间复杂度：$O(n\sqrt{k})$，空间复杂度：$O(n)$

1.26 261E Maxim and Calculator

试题大意

有a,b两个数字，1次操作可以

1.b+1

2.a=a*b

求l-r间在p次操作内可以从a=1，b=0到达的数字个数。

$l, r \leq 10^9, p \leq 100$

算法

发现l-r间用1-100间的质数可以到达数字个数$\leq 3,000,000$，用f[i][j]表示用前i个数字，拼出第j个数字的最小次数，若$f[i][j] + i \leq p$则说明第j个数字可行。

时间复杂度：$O(3,000,000*p)$，空间复杂度：$O(3,000,000)$

1.27 253E Printer

试题大意

一个打印机，每秒钟打一张纸。某些时刻它会受到一些任务，任务共有n个，每个任务有一个优先级，它要求打印k张纸。任务的优先级互不相同。打印机打纸的时候会选择优先级最高的任务。

现在只有一个任务的优先级无法知道。但是知道了打印机完成它的时刻。求一个合法的优先级。

$n \leq 50000$

算法

首先，那个任务优先级是可以二分的。

所以，先把可行的优先级列张表出来，共有O(n)个，然后在表内二分位置任务的优先级。知道优先级后，用堆来模拟。

此题还有$O(nlogn)$的复杂度的其他做法。

时间复杂度：$O(nlog^2n)$，空间复杂度：$O(n)$

1.28 309E Sheep

试题大意

每只羊有一个区间[li,ri]，两只羊若区间有交则它们之间有一条链子。你需要将它们从1-n排好序，最小化有链子的两只羊的距离。

算法

最小化问题可以用二分解决。

对于一个二分的距离mid，将羊的ri从小到大排序。羊肯定是ri越小的放越前面。但是有一些羊由于距离的限制在某些时刻不得不放入序列。现在将羊一只一只地放入，设last[i]为第i只羊必须在那个位置之前放置，sum[i] 为必须在i 之前放置的羊数。当$sum[i] > i$时必定无解了，否则找到第一个sum[k]=k的地方，说明这里有羊卡住了。然后找一个$last[j] \leq k$的羊且rj最小的羊放入。

时间复杂度：$O(n^2)$，空间复杂度：$O(n^2)$

1.29 316E3 Summer Homework

试题大意

一个整数序列a1,a2...an，有m个操作，操作有3种：

1.改变一个数字

2.计算$\sum_{x=0}^{r-l} f_x \times a_{l+x}$ 模10^9的值。

3.区间加一个数字

f为斐波那契数列。

$n, m \leq 200000$

算法

斐波那契数列有一个性质：

$F_x = F_k \times F_{x-k} + F_{k-1} \times F_{x-k-1}$

设$G(l, r, x)$为$\sum_{i=l}^{r} a_i \times F_{i+x}$

只要知道$G(l, r, x)$与$G(l, r, x+1)$的值，就可以推得$G(l, r, x+k)$了。

这样就可以用线段树维护答案了，1,3操作也可以简单的维护。

时间复杂度：$O(mlogn)$，空间复杂度：$O(n)$

1.30 243C Colorado Potato Beetle

试题大意

一个无限大的格子图上，有n条横着或者竖着的格子上有障碍。问有多少格子与无限远的地方四联通。

$n \leq 1000$

算法

将与这n条有关的坐标离散后得到大小为n*n的格子图，然后用bfs模拟一遍。

时间复杂度：$O(n^2)$，空间复杂度：$O(n^2)$

1.31 316G3 Good Substrings

试题大意

给你一个大串，和n个小串。要你求有多少不同的大串的子串满足它在第i个小串中出现次数在[li,ri]内。

串长l不超过50000，$n <= 10$

算法

用后缀自动机做：

把所有串加上分隔符并起来，加到后缀自动机中，在后缀树中统计这个点的子树中1-n的小串各有几个。在对于那些大串中的点且符合答案的加入答案。

时间复杂度：$O(n^2 * 50000)$，空间复杂度：$O(n * 50000)$

1.32 251E Tree and Table

试题大意

把一棵2n个点的树塞进一张2*n的表中。表中每个格子表示树的一个节点。树中相邻的点在表中也要相邻。求方案数。

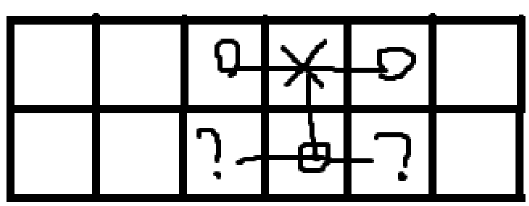
$n \leq 100000$

算法

如果n=1，答案为2

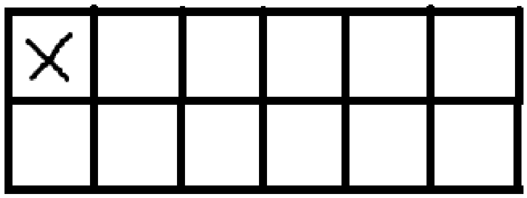
如果树是一条链，则答案可以手推得到。

否则树中一定有一个点度数为3。很显然它在表中占据了中间的位置，且与之相邻的3个点分别放进了表中相邻的3个格子中。如下图所示，这里可以连同问号的对应点枚举每一种情况。在这里这些点具体在表中的哪些位置是不需要知道的，只要确定它们的相对位置就可以了。就是说只要左右两端没有单个的，且中间全部都填满了，那最终它们在表中的位置也就知道了。



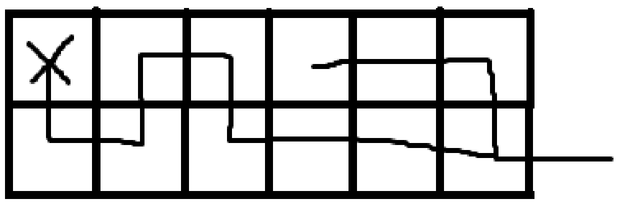
先定义$G(x, y)$为x节点与y节点分别在2行的表的最左端的匹配方案数，比如上图的X右半部分。那肯定是x与y一开始都是一条长长的链，之后有一个用完了所有的节点，变成了$F(x)$。

再定义$F(x)$为x节点在2行的表的最左端的匹配方案数。如图：

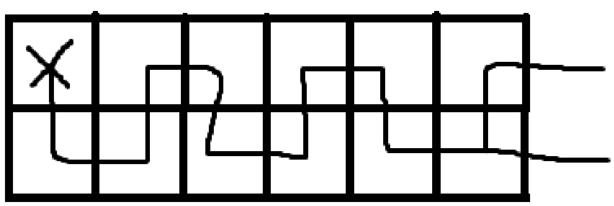


考虑它的转移，先找到x子树中离x最近的拥有2个儿子的节点y。

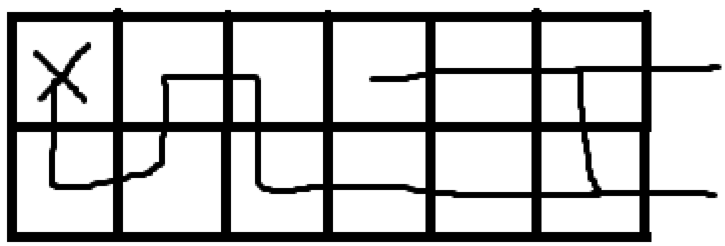
转移1：如果这时y的一个儿子是一条链，则可以用这条链和x到y的链去覆盖前面的格子，比如这样：



转移2：如果x到y的链已经全部覆盖，那y的2个儿子可以去覆盖之后的格子，比如这样：



转移3：比较难想到。如果y的一个儿子拥有一条链与一棵子树，则这个儿子的链可以与x到y的链覆盖前面的，比如这样：



可以证明没有其他情况了。

用dp求F与G就可以了。

时间复杂度：$O(n)$，空间复杂度：$O(n)$

1.33 256D Liars and Serge

试题大意

有n个人，每个人要么说谎要么说真话。

现在有人问他们:你们中有几位总是说真话呢？诚实的人总是回答正确的答案，而说谎的人会回答1到n之间除了正确答案以外的任意一个整数。每个说谎的人都会选择他自己的答案，并不会考虑其他说谎的人的答案，所以两个说谎的人可能会给出不一样的答案。他拿来一张纸并写下n个整

数a1,a2,...,an，其中ai表示第i个人的答案。

现在问有多少种不同的a序列使得你可以看出有恰好k个人明显说谎了。

$1 \leq k \leq n \leq 2^8$，保证n是2的幂次。

算法

假如说真话的人有k个，那么只有当 $ai = k$ 的个数恰好是k个的时候他们才有可能说的是真话，否则就是明显说谎。

因此采用如下dp：$f[i][j][k]$ 表示做到i这个数字，已经有j个人明显说谎了，a序列中已经填了k个数字的方案数。转移时枚举有l个a等于i这个数字，如果$l \neq i$，则这l个人都在明显说谎，再乘一个组合数转移。

但是这样复杂度是n^4的，在1s内无法通过。但是题目保证n是2的幂次，情况数其实很少，所以可以用打表的方法。

时间复杂度：$O(n^4)$，空间复杂度：$O(n^3)$

1.34 303D Rotatable Number

试题大意

如果在b进制下，一个长度为n的数字x满足：x*1,x*2...x*n这些数字与x的n种通过旋转得到的数字相同，则称x在b进制下是一个循环数。允许

有前导0。下面是一个例子：

$$142857 * 1 = 142857;$$
$$142857 * 2 = 285714;$$
$$142857 * 3 = 428571;$$
$$142857 * 4 = 571428;$$
$$142857 * 5 = 714285;$$
$$142857 * 6 = 857142$$

给出n，x。求最大的b($1 < b < x$)，满足存在一个长为n的在b进制下的循环数。

$1 \leq n \leq 5 \times 10^6, 2 \leq x \leq 10^9$

算法

将循环数看成一个小数的循环节部分，比如142857看成$0.\dot{1}4285\dot{7}$。设这个数是$\frac{1}{p}$，通过Midy's theorem可以证得$p = n + 1$。

现在反过来，考虑一个循环小数$\frac{1}{p}$要满足什么条件，它的循环部分才可以当做循环数。考虑b 进制下的除法过程，循环小数的第i位为$\lfloor \frac{b^i mod\, p}{p} \rfloor$。$b^i\ mod\, p$形成了一个环状的结构，而如果要满足$\frac{1}{p}$，$\frac{2}{p}...\frac{p-1}{p}$满足循环数的条件，就要满足在i在$[1, p-1]$内，$b^i\ mod\, p$互不相同（因为$\frac{x}{p}$相当与从x开始进入环状结构）。从中得到p 与b互质，p是质数，且b是p的一个原根。

回到原题，由于b要是n+1的原根，所以从大到小只要枚举最多n+1个b进行验证就行了。

要特判n=1的情况。

时间复杂度：$O(nlogn)$，空间复杂度：$O(logn)$

1.35 338D GCD Table

试题大意

有一张n*m表A，A(x,y)=gcd(x,y)

现有k个数字，求这些数字是否连续出现在表A的某一行中。

$n, m \leq 10^{12}, k \leq 10^4$

算法

先将k个数字的lcm求出来，设等于v。则v必定是最优的行号。（即使v的倍数也可行但是不优）接下去就要用列去匹配。

首先，如果两个数字ai,aj相隔s，但是gcd(ai,aj)!=s，这肯定有问题。

设匹配中的第一位是第x列，那么可以得到

$$(x+1)\ mod\ a[1] = 0$$
$$(x+2)\ mod\ a[2] = 0$$
$$...$$
$$(x+k)\ mod\ a[k] = 0$$

此为线性剩余方程组，只要对于v的每一个质因子，求出最大的那个，做中国剩余定理即可。

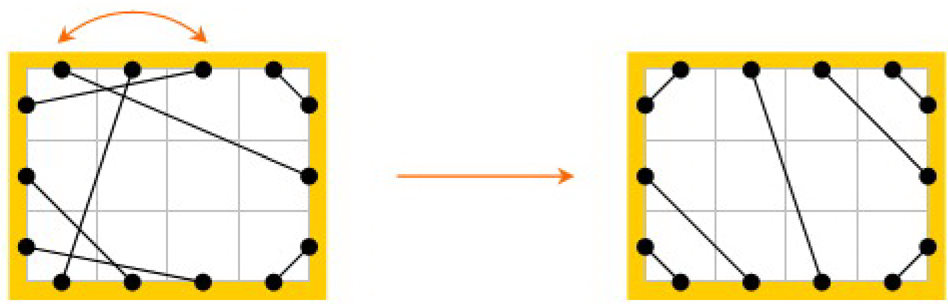
时间复杂度：$O(klogn)$，空间复杂度：$O(k)$

1.36 269E String Theory

试题大意

一个n*m的网格图，边的中点上有一个节点。有n+m条线，每条线连接两个节点。这两个节点不会再一条边上。

如图：

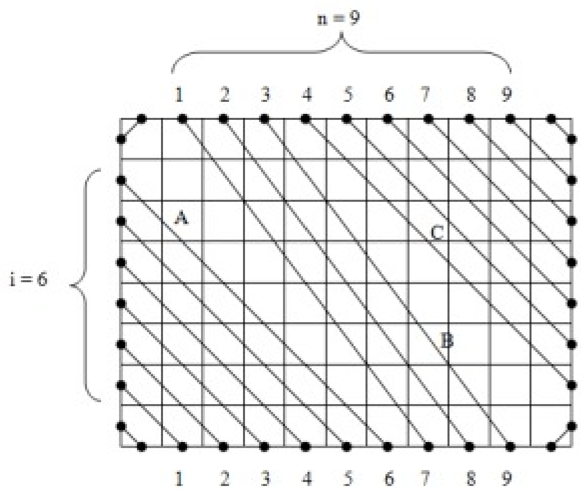


现在可以交换两行或者两列，行/列上的节点也会相应调换。求一种行和列的排列方案使得没有两条线会交叉。

$n, m \leq 10^5$

算法

如果没有两条线交叉，最终的形态是可以确定的，如图：



把线分为两类，一类是四个角落上的，还有一类是中间的。

为了方便，假设每行两个端点连上线，每列的也类似。

先考虑角落上的：

角落上的肯定是一些长度为8的环。在原图中找到这些环后依次放入即可。

中间的就比较麻烦了，如上图所示。不妨假设n小于m，给横排的每个点编上编号后，可以看出上排编号为k 的点沿着边走，会走到下排编号为$((k+i-1) mod m + 1)$号点。是一个差为i的循环。可以得出共有gcd(i,n)个环（此处n为图中的n），若将线划分成连接上下，连接左上等等之类的，则第i个环从上排i号点开始走经过的边的模式都是相同的。因此在原图中找到这些环，然后做一下边模式的循环匹配即可。可以用kmp实现。

时间复杂度：$O(n+m)$，空间复杂度：$O(n+m)$

1.37 332E Binary Key

试题大意

假设有p和q两个长度为正整数的字符串，我们分别叫它们为匣子和钥

匙。

其中，钥匙串q只包含字符0和1。

下面给出一段简单的算法用以描述如何从匣子串p中提取出信息串s

```
i = 0;
j = 0;
s = <>;
while i is less than the length of the string p
{
if q[j] == 1, then add to the right of string s character p[i];
increase variables i, j by one;
if the value of the variable j equals the length of the string q, then j = 0;
}
```

我们知道这个算法实现起来非常简单，所以你的任务会稍微复杂一些。你需要构造一个字典序最小的长度为k的钥匙串。使该钥匙串用于以上算法时，从给定的匣子串p中获得给定的信息串s

$|p| \le 10^6$, $|s| \le 200$, $k \le 2000$

算法

q中1的个数，可以用s的长度与k推出最多只可能有2种可能。

假设知道了q中1的个数后，事情就变得很简单：s的间隔与p的间隔都知道了，s从头到尾匹配一下就行了。

时间复杂度：$O(|p|)$，空间复杂度：$O(|p|)$

1.38 331E2 Deja Vu

试题大意

一个n个点m条边的有向图，没有两条边连接了同一对点，你经过一条边的时候，会看到一些点的编号。

一条路径是好的，当且仅当，这条路径上的点构成的序列，和这条路径上的边上的编号构成的序列完全一样。

要求统计长度为1,2,...,2*n的好路径的个数。答案对10^9+7取模。

算法

如果确定了路径上的一条边，且这条边上的编号有很多，那么能找到路径上的一系列的点。

根据这个想法，把一条好路径分成若干条链，每一条链都是以一个点出现在一条边的编号中的这些边为中心。这些链有些是左闭右开的，或者左开右闭的...怎么求这些链呢？只要枚举那些中心边，枚举这个点出现在哪里（注意：这个点可能会在边上的编号序列中出现多次）。将这些边往前往后扩展到不能扩展为止。这之后只要做一次dp 就行了。

时间复杂度：$O(n^2m+n^4)$，空间复杂度：$O(n^3+nm)$

1.39 356E Xenia and String Problem

试题大意

如果一个串s是gray的，要满足3个条件：

1.长度是奇数

2.$s[\frac{|s|+1}{2}]$ 在s中出现了1次。

3.要么 $|s|$ 是1，要么 $s[1..\frac{|s|+1}{2}-1]$ 和 $s[\frac{|s|+1}{2}+1..|s|]$ 相同且都是gray的。

一个字符串的价值为所有是gray的子串的长度平方之和。

现在给你一个长n的串，你可以改变其中一个字符，求最大价值。

串由小写字母构成。

长度 $N \leq 100000$

算法

如果没有修改的话，算出每个位置为中心的最长的gary串，记长为f[i]，串的价值为 $\sum g[i]$ ，

$g[i] = 1^2 + 3^2 + 7^2 + ... + f[i]^2$

若我们修改了i这个位置的字符，则那些包含i位置且i不是中心的gary串都不再是gary串了。那些原来正是因为i不和谐而无法成为gray串的变成了gray串。

前者我们可以用前缀和算出包含i的价值总和，后者考虑一个不是gary串变成了gary串，肯定是因为左右两边有一边包含了i，而i改变后变成gary串了。因此可以逆向从i 位置暴搜出去。因为每个从不是gray 串到是gray 串的那些串，都会被唯一的搜到。因此搜索的复杂度也是有保证的。

时间复杂度：$O(26nlogn)$，空间复杂度：$O(nlogn + 26n)$

1.40 273E Dima and Game

试题大意

Dima和Anya喜欢玩各种游戏。现在Dima想出了一个新游戏想和Anya玩。

Dima在纸上写下n对整数$(l[i], r[i])(1 <= l[i] < r[i] <= p)$。然后玩家轮流进行操作。轮到自己时，可以进行下面的操作：

1.选择第i对数$(1 <= i <= n)$，满足$r[i] - l[i] > 2$;

2.将第i对数替换为$(l[i] + floor((r[i] - l[i])/3), l[i] + 2 * floor((r[i] - l[i])/3))$或者$(l[i], r[i] - floor((r[i] - l[i])/3))$。$floor(x)$表示向下取整。

不能进行操作的玩家则输。

当然，Dima希望先进行操作的Anya赢得游戏。所以Dima需要写下这样的n对整数$(l[i], r[i])(1 <= l[i] < r[i] <= p)$，使得如果两个玩家都采取最

优策略，先操作的玩家取得胜利。请计算Dima有多少种这样的方法。输出方案数模1000000007(10^9+7)后的值。

如果这些数对被按照不同的顺序写了下来，则将其视作不同的方法。

算法

(l,r)的sg值与$(l+1,r+1)$的sg值是一样的，这个从转移中可以看出，所以sg值只与$r-l$有关。如果在暴力求sg的时候将相同sg的压缩在一起当做一个块，发现两种转移都在同一个块内的数是一个个区间。如果将这些数字一并做掉合成一个块的话，最终的块数只有100块左右。然后算出每种sg的(l,r)对数。最终用dp求解。

时间复杂度：$O(100+n)$，空间复杂度：$O(100+n)$

1.41 333C Lucky Tickets

试题大意

如果一个在数字的个数码左右放置运算符号及括号，使得运算结果为k的话，此数字被称为k幸运数。给出n，要求出m个不同的8位的n幸运数。

$N <= 10^4, m <= 3*10^5$

算法

一种构造方法：前四位乱运算，后四位去凑到n。

时间复杂度：$O(m)$，空间复杂度：$O(1)$

1.42 240F Torcoder

试题大意

给出一个长度为N的字符串，以及M个操作。每个操作给出一个区间[l,r]，你需要将[l,r]中的字符重排，使得这个子串成为一个回文串——如果有多种方案，你要使得字典序最小。每个操作结束后都不撤销，也就是将字符串依次进行M次变换。如果某个操作不可能排出回文串，则直接无视这个操作。

你需要输出经过M次操作后的字符串。

算法

由于是变成字典序最小的回文串，所以每种字符都是头尾两段。直接维护26棵线段树，第i棵线段树用来记录哪些位置上有字符i。只要执行区间覆盖，区间求和即可。

时间复杂度：$O(26nlogn)$，空间复杂度：$O(26n)$

1.43 305E Playing with String

试题大意

一开始有一个字符串s。

每次操作可以选择一个字符串t，然后选择一个i，满足$t[i-1] = t[i+1]$，然后将t分裂成$t[1..i-1], t[i..i], t[i+1..|t|]$。

求先手是否必胜与第一步操作。

$|s| \leq 5000$

算法

直接求sg是$O(n^3)$的，需要优化。

如果$s[i-1]=s[i+1]$，那么i-1向i+1连一条边。把相交的边称作一块，不同块之间是互相没有影响的。同一块之间边肯定是连续的一段。所以可以直接用1维状态——边数，来表示一块的sg。

时间复杂度：$O(|S|^2)$，空间复杂度：$O(|S|)$

1.44 235D Graph Game

试题大意

Solve(t)

(t是一张n个点，n条边的连通无向图)

1.totalCost=totalCost+(size of T).运算符’=’表示赋值。(Size of T)表示图T中的结点个数。

2.在图T中随机选择一个结点x(图T中每个点被选中的概率相等)

3.从图T中删除结点x

4.然后T变成了一些联通快

5.分治处理所有的Solve(S) (S是剩下的连通块)

求totalcost的期望值。

$N \leq 3000$

算法

考虑一棵树上的时候，total的期望$=\sum f(a,b)$，f(a,b)表示的是在a删除时，a与b连通的概率。设在树上a与b间的点数为n，只要a删的比所有点都早。所以f(a,b)=1/a与b之间的点数。考虑n条边的情况，a 与b 有两条路径。容斥一下。

时间复杂度：$O(n^2)$**，空间复杂度：**$O(n)$

1.45 342D Xenia and Dominoes

试题大意

一个3*n的格子图，里面有一些障碍物。你要用一些1*2的多米诺骨牌覆盖这个格子图。格子图中有一个空白点，他不能被覆盖。骨牌在空白处可以移动。横向的多米诺骨牌只能横向移动，纵向的多米诺骨牌只能纵向移动。求至少可以移动一次的覆盖方案数。

$n \leq 10000$

算法

状态压缩dp，维护分割线处的状态。做到空白点附近时看一下横向和纵向的骨牌能否让它移动即可。

时间复杂度：$O(n)$**，空间复杂度：**$O(n)$

1.46 319E Ping-Pong

试题大意

如果区间(a,b)能走到(c,d)，要保证$c < a < d$ 或$c < b < d$

支持2个操作：

1. 添加(x, y) 保证$y - x$ 严格递增

2. 询问第x次加入的能不能走到第y次的。

操作数≤ 100000

算法

用并查集维护能互相到达的区间，由于区间长度递增，所以用线段树记录所有包含[l,r]的区间，插入新区间时判断端点即可。

同一个并查集内的区间可以合成一个大区间询问时只要第y个区间是否在x并查集的大区间内就可以了。

时间复杂度：$O(n\ logn\ \alpha(n))$，空间复杂度：$O(n\ logn)$

1.47　301C Yaroslav and Algorithm

试题大意

Yaroslav喜欢算法。我们将描述一个他最喜欢的算法。

1.这个算法接受一个字符串作为输入。我们设这个输入字符串为a。

2.这个算法由一些命令组成。i号命令的形式为”s[i]¿¿w[i]”或”s[i]¡¿w[i]”，其中s[i]和w[i]是长度不超过7的字符串（可以为空），由数字或字符”?”组成。

3.这个算法每次寻找一个编号最小的命令i，使得s[i]是a的子串。如果没有找到这样的命令，那么整个算法终止。

4.设找到的命令编号为k。在字符串a中，s[k]第一次出现的位置会被w[k]替换。如果这个命令形如”s[k]¿¿w[k]”，那么这个算法继续执行（译注：回到第3步）。否则，算法终止。

5.算法的输出就是算法终止时字符串a的值。

Yaroslav有一个n个正整数的集合，他需要一个这样的算法，且能够使每一个数加1。更正式地，如果我们把每个数看成一个十进制表示的字符串，那么对于每个字符串独立地运行这个算法，这个算法需要输出一个输入串对应的数+1的字符串。

算法

这道题只要设计出一个能输入n输出n+1的算法，就与本题输入无关了。

下面有一个不错的算法：

```
??0>>0??
??1>>1??
??2>>2??
??3>>3??
??4>>4??
??5>>5??
??6>>6??
??7>>7??
??8>>8??
??9>>9??
??>>?
0?<>1
1?<>2
2?<>3
3?<>4
4?<>5
5?<>6
6?<>7
7?<>8
8?<>9
9?>>?0
?<>1
>>??
```

这段代码的意思就是先塞一个 ?? 到最末尾，然后将 ?? 变成 ?，后来一碰到 x?就变成$x+1$?。x=9的时候判一下。

时间复杂度：$O(1)$，空间复杂度：$O(1)$

1.48 309D Tennis Rackets

试题大意

一个边长为n+1的正三角形。每条边上除去端点后，都很整齐的放着n个间隔为1的小孔。只有距离两个端点超过m的孔是有用的。现在要每边选一个孔，另这三个孔组成的三角形为一个钝角三角形，求方案数。

$n \leq 32000$

算法

看似n很大，其实就是很简单的暴力。枚举钝角的一边，再枚举另一边，这时可以算出另一边点的区间，然后累加一下。需要适当的优化才能过。

时间复杂度：$O(n^2)$，空间复杂度：$O(n)$

1.49 301E Yaroslav and Arrangements

试题大意

称一个序列$a[1..r]$是good的，要满足

1.$a_1 - a_2 = 1$, $|a_2 - a_3| = 1, \dots, |a_r - a_1| = 1$

2.$a_1 = min(a_{1..r})$

称一个序列$b[1..e]$是great的，要满足：

1.非递减

$2.1 \le r \le nand1 \le b_i \le m$

3.乱排列后能得到至少1个至多k个的不同的good序列。

给出n,m,k求不同的great序列个数。

$n, m, k \le\ 100$

算法

显然是dp，但如何表示状态？

1.b元素个数显然要记(i)

2.假设数字从1开始放，现在放到了第几个数字(j)

想象把b序列的数字一个个插入a中，那么当前的数字能放入的位置只和上一个数字留下来的槽的数量有关。那么又有两个状态：

3.已经得到的未填满的good序列数(k)

4.上一次留下的槽数。(l)

转移：枚举现在比留下的槽数多x，方法数变成$k * C_{(l+x-1)}^{(l-1)}$ （l+x 个球放入l个盒子，不能有空）。

虽然复杂度是$O(n^4k)$ 但因为有用的状态不多所以可以过。

内存需要滚存才能通过。

时间复杂度：$O(n^4k)$，空间复杂度：$O(n^3)$

1.50 309B Context Advertising

试题大意

给出一段文字，要将连续一段塞进r行的文本，每一行最多只能有c个字符。要求最大化单词数。长度$N \le 5 \times 10^6$

算法

假如知道了某一行是从第i个单词开始，那一定是贪心地尽可能放多的

单词进去，直到这一行满出来了再放进下一行中。假如从第i个单词开始放能一直放到j，就从i向$j+1$连一条边。如果从i 开始顺着边跳r次，就是r行的答案了。

这些边其实构成了一棵树，顺着边跳r次相当于树上距离自己r的祖先。于是dfs一下顺便把祖先的信息用一个栈来维护就可以了。

时间复杂度：$O(N)$，空间复杂度：$O(N)$

1.51 331D3 Escaping on Beaveractor

试题大意

$m*m$的平面上有n个平行于坐标轴的箭头，一个箭头是一条有方向的线段。一个人在这个平面上走，走到箭头后方向就会变成它的方向了。

现在有Q个询问，每次给定初始点与一个方向，以每秒1单位的速度移动，问移动t_i秒后在哪里。

$n,m,Q \leq 10^5$, $t_i \leq 10^{15}$

算法

可以用扫描线+线段树预处理出从每条箭头的末端或者询问的点出发，向某个方向前进第一个碰到的箭头是哪个。每个箭头当成一个点，向从它出发第一个碰到的箭头连一条边，这就是一个环套外向树的结构。

每次询问只要在这个结构上倍增即可。

时间复杂度：$O((n+q)logm\ +\ qlog\ t_i)$**，空间复杂度：**$O(n+q+m)$

1.52 277D Google Code Jam

试题大意

有n道大题，每道大题有两小题，做对的得分分别为 s_{1i}, s_{2i}，所需的时间分别为 t_{1i}, t_{2i}。第一小题一定能做对，第二小题必须要完成第一小题后才能做，且第二小题有 p_i 的概率做错。现在有 t 时间来做这些题。做题目的总用时为正确做完的最后一小题的完成时间。求期望总分的最大值，和满足期望总分最大的前提下，最少的期望用时。

$n \le 1000$, $t \le 1560$

算法

假设做的题目的集合已确定，如何安排顺序使得总用时最少。

首先显然是先把全部的第一小题安排在前面，所以只考虑第二小题的排列。假设第二小题的最优排列中，题目 i 和 j 是相邻的，它们的贡献是

$$(S+t_{2i})p_j(1-p_i)P+(S+t_{2i}+t_{2j})(1-p_j)P$$

其中 S, P 都是常数。现在交换 i, j，要求现在的贡献大于刚才的贡献，消去常数，即

$$\begin{gathered}
t_{2i}p_j(1-p_i)+(t_{2i}+t_{2j})(1-p_j) < t_{2j}p_i(1-p_j)+(t_{2i}+t_{2j})(1-p_i) \Longrightarrow \\
-t_{2i}p_ip_j - t_{2j}p_j < -t_{2j}p_ip_j - t_{2i}p_i \Longrightarrow \\
t_{2i}p_i(1-p_j) < t_{2j}p_j(1-p_i)
\end{gathered}$$

所以把每道大题按上面的规则排序，然后直接动态规划即可。

时间复杂度：$O(nt)$，空间复杂度：$O(t)$

1.53 341E Candies Game

试题大意

n个盒子，每个盒子里有一些糖a_i。有一个操作：选择两个盒子i,j，假如$a_i \leq a_j$，然后从j盒子里拿a_i个糖放进i盒子。

游戏的目的是另恰好两个盒子里的糖数不为0。

$n \leq 1000, \sum a_i \leq 10^6$

算法

假设三个盒子里的糖分别为x, y, z，假设$x \leq y \leq z$

现在要通过一系列操作使得y最终变成$y \; mod \; x$，可以知道在log步左右就可以将一个盒子里的糖数变为0。n个盒子的游戏也只要三个三个玩就行了。

流程如下：另$p = \lfloor y/x \rfloor$

```
while(p){
        if(p&1)y-=x,x+=x;
        else z-=x,x+=x;
        p/=2;
}
```

类似于二进制分解。由于$z \geq y$，因此上述算法是对的。

时间复杂度：$O(nlog\ t_i)$**，空间复杂度：**$O(nlog\ t_i)$

1.54 323B Tournament-graph

试题大意

你要构造一个有N个结点的竞赛图，使得对任意两个结点u和v（u≠v），从u到v的最短距离不超过2。

竞赛图就是基图为无向完全图的有向图（每对结点之间有一条有向边相连，且无自环）。

$n \leq 1000$

算法

使用增量法。如果已经构造好了n个点的图，可以让所有点向n+1号点连边，n+2号点向所有边连边，n+1号点向n+2号点连边。这样就构造好了n+2个点的图。

如果n是奇数，从n=1开始即可。

如果n是偶数，n=4是无解的，手造一个n=6的解再推下去就可以了。

时间复杂度：$O(n^2)$**，空间复杂度：**$O(n^2)$

1.55 248E Piglet's Birthday

试题大意

N个柜子，每个柜子有a_i个糖罐。M个操作，每次操作从i这个柜子随便选k 个罐子，尝一下在放到j柜子。问每个操作后全是尝过的罐子的柜子数的期望值。

$n \leq 10^5$, $a_i \leq 100$, $m \leq 10^5$, $k \leq 5$

算法

可以将操作分开，变成两个操作：

1.一个柜子去掉k个罐子

2.一个柜子加上k个尝过的罐子。

只要维护每个柜子还有j个罐子没尝过的概率就行了。

时间复杂度：$O(kMax(a_i)M)$，空间复杂度：$O(Max(a_i)N)$

1.56 325C Monsters and Diamonds

试题大意

Piegirl 发现一只怪物和一本关于怪物和馅饼的书。当她在读这本书的时候，她发现有n 种怪物，每种都有一个唯一的1 到n 的编号。如果你喂怪物一块饼，它就会分成一定数量的怪物（可能为零），以及至少一个多彩的钻石。怪物们可能存在多种分裂方式。

最开始Piegirl 有且只有一只怪物。她先喂了一块饼，它随之分裂。对于分出来的怪物，继续喂饼，直到它们都分裂为钻石。然后她把所有钻石收集起来。

你将得到一系列规则描述不同怪物的分裂方式，每种怪物至少有一种分裂方式。分裂的时候，如果有多种方式，Piegirl 可以选择任意一种。

你的任务是：对于每种怪物，确定以其为起始，Piegirl 可以得到的钻石最少最多分别是多少。Piegirl 有无限多的饼。

如果没办法将怪物都分为钻石，则输出-1 -1。否则如果能分完所有怪物且得到无穷多的钻石，用-2 作为无穷大。

$n, m \leq 10^5$

算法

先用拓扑序算出从每个怪物开始能否最终都变成钻石，如果可以的话

可以顺便用优先队列求出最少能变成多少个。然后把不能的怪物全部去掉，再dfs一遍求出最多能变成多少钻石就可以了。

时间复杂度：$O(n + mlogn)$**，空间复杂度：**$O(n + m)$

1.57 266D BerDonalds

试题大意

给定一个n个点的无向带权联通图，求图的直径。

$n \leq 200$

算法

枚举每一条边，假设重心离其中一个端点的距离为x，则其他点到重心的距离是一个关于x的分段函数，而且形状是45°向上再45°向下。然后求出最优点就行了。

时间复杂度：$O(n^3logn)$**，空间复杂度：**$O(n^2)$

1.58 311C Fetch the Treasure

试题大意

用1-h个房间，你的出发点是1，一开始你每次只能前进k个房间。有n个房间有宝藏，宝藏有一定价值。

有m个操作，每个操作分三种：

1、允许你每次前进x个房间。

2、让第x个房间的宝藏价值减少y。

3、询问你能到达的房间中价值最大的宝藏。

1号操作不超过20个。

$h \le 10^{18}, n, m \le 10^5, k \le 10^4$

算法

如果你能走到第x个房间，那么$x+k, x+2k...$都可以走到，用f[x]表示模k等于x的房间中能走到的最小房间编号。每一个1操作后都可以用最短路解决。

第2个操作可以用一个堆来解决。

时间复杂度：$O(mlogn + 20klogk)$，空间复杂度：$O(n+k)$

1.59 316D PE lesson

试题大意

n个人，每个人拿了一个球，球互不相同。每次可以选两个人交换他们手中的球。每个人的交换次数不能超过2次。问最终人手中的球有几种可能的排列。

$n \le 10^6$

算法

考虑最终的排列，每个人向自己手里的球的初始拥有者连一条边，会连出一些环。如果这个环中交换1次的人不超过2个，则这个环是可以交换出来的。

假设交换1次的有n个，交换2次的有m个。

可以先用dp算出n个交换1次的分配到若干环中的方案数，剩下的能交换2次的，每个人都能跟到前面任意一个人的后面。答案就是：

$f[i] = f[i-1] + f[i-2] * (i-1)$

$answer = f[n] \times \frac{(n+m)!}{n!}$

时间复杂度：$O(n)$**，空间复杂度：**$O(1)$

1.60 323C Two permutations

试题大意

你有两个各包含n个元素的排列p和q，和m个由l1，r1，l2，r2组成的询问。每次询问既在p[l1,r1],又在q[l2,r2]中的数的数量。

$n \le 10^6$, $m \le 200000$

算法

函数式线段树

时间复杂度：$O((n+m)logn)$**，空间复杂度：**$O(nlogn)$

1.61 343E Pumping Stations

试题大意

一张n个点m条边的无向图G，每条边有流量。要求一个排列v,使得最大化$\sum_{i=1}^{n-1} Maxflow(G, s = v_i, t = v_{i+1})$

$n \le 200, m \le 1000$

算法

最大流=最小割

先做n-1遍网络流求出n个点两两之间的最小割。然后暴力建出最小割树，两个点的最小割等于最小割树上两个点路径上边权的最小值。

要求排列的话只要向找出边权最小的边，然后将树割开递归两边即可。可以证明这样是最优的。

时间复杂度：$O(n \times Maxflow(G) + n^3)$，**空间复杂度**：$O(n^2 + m)$

1.62 267C Berland Traffic

试题大意

现有一个n个点m条边的网络。每条边有一个容量限制，对于一条边$< x, y >$，设其流量为t，容量为c，如果t 大于0则是从x流向y，反之就是从y流向x，但t的绝对值不能超过c且可以不是整数。网络中节点1是源点，节点n是汇点，对于除1,n外的所有点，流入的流量等于流出的流量。这个网络有一个奇怪的性质，对于任意一对节点(x,y)，x到y的路径上流量之和不会随着选择不同的路径而改变（有可能有小于0的流量，流量的符号取决于x到y路径上这条边的方向）。

求满足上述条件且流过该网络的流量尽可能大的方案以及最大的流量是多少。

$n \leq 100$, $m \leq 5000$

算法

由于题目中要求x到y每条路径的流量和都相同，所以可以设1到x的流量和为x_i，然后根据流量平衡可以得到n-2个方程，再固定$x_1 = 0$，这样就有了n个未知数n-1个方程。高斯消元后所有未知数都表示成了$a_i * x_n$的形式，根据容量限制求出最大可能的x_n即可。

时间复杂度：$O(n^3)$**，空间复杂度：**$O(n^2+m)$

1.63 240E Road Repairs

试题大意

边权只有0,1的最小树形图，要求方案。

$n,m \le 10^5$

算法

朱刘算法。求方案的话要把每次更新后的边都存下来，这样才能很好地处理一条边替换了另一条边的操作。

据说时间复杂度是$O(mlogn)$的？

时间复杂度：$O(mlogn)$**，空间复杂度：**$O(mlogn)$

1.64 329E Evil

试题大意

笛卡尔坐标系中有n个点，两个点之间的距离等于他们的manhattan距离。求一个哈密尔顿回路使得经过的路程最大。

$N <= 10^5$

算法

不妨设没有两个点x，y坐标相同。（如果相同可以进行细小的移动）。

把x，y坐标分开考虑，答案的上界必定是：将x排序后最大的n/2取正，最小的n/2个取负。Y也如此。

假设这些点被x坐标的中位数线与y坐标的中位数线分成ABCD四个集合。

N为偶数时：$|A| = |C|, |B| = |D|$

N为奇数时：有两种可能，要么是1个点在原点，要么是2个点各自在轴上。

以此分情况贪心。

时间复杂度：$O(nlogn)$，空间复杂度：$O(n)$

1.65 303E Random Ranking

试题大意

现在请你假想一次考试，这场考试有N个考生。

你可以预测这N个考生最终得分的范围，从而预测他的排名。选手们的分数将按照成绩升序排列，也就是说，分数越高，排名越后。

你知道第i个考生的得分范围是[Li,Ri]（得分可以是实数），现在你要预测每个考生得到任何排名的概率。

$n \leq 80$

算法

离散出$O(n)$个分数段，对于第i个人来说，枚举他最终得分在哪一段中，别的人有一定概率得分在比这一段低的分数段中，也有可能出现在同样的分数段中。所以用$f[k][x][y]$表示对于前k个人来说，有x个人分数一定比i低，y个人分数段与i一样的概率。如果有y个人分数段与i一样，那i排在$1..y+1$名的概率都是$1/(y+1)$。

时间复杂度：$O(n^5)$，空间复杂度：$O(n^2)$

1.66 316F3 Suns and Rays

试题大意

一张h*w的01图片中有一些太阳，太阳有一些光线。你的任务是计算太阳的数量和计算每个太阳射线的数量。

太阳是带有光线的可以任意旋转的椭圆。射线是一条连接在椭圆边界上的线段。

没有两个太阳有共同点。

射线的宽度为3像素。

太阳的轴的长度将介于40和200像素。

没有两条射线相交。

所有射线的长度将在10和30像素。

$h, w \leq 1600$

算法

以每个点为中心做一个9*9的正方形，如果有太阳的部位将其分成了两个较大的空白，则认定这个中心是射线的一部分。

这样处理完以后再在每个太阳上dfs一下数一下有几个较大的射线部分的联通块就行了。

时间复杂度：$O(nm)$，空间复杂度：$O(nm)$

1.67 360E Levko and Game

试题大意

N个点，m条边，有额外的k条边，这些边的权值可以在li ∼ ri中任意挑选。现在你在s1，对手在s2，你可以设置k条边的权值，如果你到t的距离小于他到t的距离你就赢了。不能赢就尽量平手。

$N, m \le 10000, k \le 100$

算法

先做最短路。如果一条边$a \to b$，边权$l \sim r$，考虑赢的情况，如果s1到a的距离<s2到a的距离，那就把边的权值改成l。(更多可能的让s1到其他点的距离<s2)。就这样不停地做这个操作直到没有边被修改位为止。

平手的情况，只要把<改成<=即可。

时间复杂度：$O(kmlogn)$，空间复杂度：$O(n+m)$

1.68 338E Optimize!

试题大意

n个数$a[1..n]$，m个数$b[1..m]$。求有多少个x满足：$a[x..x+m-1]$与$b[1..m]$存在一种两两配对的方式使得每一对的和都大于h。

$n, m \le 150000 ai, bi, h \le 10^9$

算法

将b从小到大排序，ai可以与一段b的后缀匹配。假设ai能匹配$b[p_i..m]$。

如果$\{p_x, p_{x+1}, \ldots, p_{x+m-1}\}$满足$\forall j \in N^*, 1 \le j \le n$,有$(\sum_{1\le k\le n, p_k\le j} 1) \ge j$，则这是合法的。

这个问题很经典，只要维护一颗线段树就可以判断了。从$a[x..x+m-1]$移到$a[x+1..x+m]$时，只要加减一个数字即可。

时间复杂度：$O((n+m)logm)$，空间复杂度：$O(n+m)$

1.69 285E Positions in Permutations

试题大意

P是n个互不相同且不超过n的正整数的一个排列。我们设排列P的第i个元素为P[i]，n为排列的长度。

我们称排列中的第i个位置是完美的，当且仅当$|P[i]-i|=1$。

请求出长度为n的而且完美的位置数刚好为k的排列数是多少。答案要求取模10^9+7。

$k \le n \le 1000$

算法

可以用dp求出完美的位置数至少为k的排列数是多少，然后用容斥即可。

时间复杂度：$O(n^2)$，空间复杂度：$O(n^2)$

1.70 238D Tape Programming

试题大意

有一个由. $<$ .. $>$ ..0.,.9.组成的序列，现在有程序(cp为指向序列的指针，dp为cp移动的方向)：

1.如果cp指的是数字，那么输出这个数，并且这个数字减1,如果这个数字减到负了就删掉。

2.如果cp指的是. $<$.或. $>$.，cp上一个指的也是这个，那么把上一个删掉。dp就变成对应的。最后cp前进一步。

m个询问，每次把li-ri的拎出来单独做，cp在最左端，dp指向右。问你0-9各出现了几次。

序列长度$n, m \leq 100000$

算法

先全部做一遍，执行次数不会超过10n。把所有的操作都记下来。每次询问相当于询问在第一次cp进入li时，到cp第一次走出li或者走出ri之间的所有输出。

考虑以下数据：

$> 8 < 8 < 8 < 8 < 8 < \cdots < 8$

模拟时暴力做是不行的。要用链表。

询问时离线+树状数组。

时间复杂度：$O(10n + mlogn)$，空间复杂度：$O(100n)$

1.71 331C3 The Great Julya Calendar

试题大意

一个数字n，每次取一个数位上的数x，然后n减掉x。用最少的步数让n减到0。

$n \leq 10^{18}$

算法

首先，每次取最大的数字减掉肯定是最优的。

为了方便理解，先设$F(x,y)$为：将x和y连接起来这个数字将y减到0以下的最少步数。如$F(123,456)$就是将123456减到123000以下的最少步数。可以发现x对步数的贡献只有体现在x数位最大值上，所以就可以优化成$F(x,y)$为：前面数位最大的是x，将y减到0以下的最少步数。这个是可以递归计算的。由于y的取值只可能是$10^k-9\sim 10^k$ 或者n的后几位，所以状态不会太多。

时间复杂度：$O(10^3 * logn)$，空间复杂度：$O(10^2 logn)$

1.72 288E Polo the Penguin and Lucky Numbers

试题大意

给出两个只由 4,7 构成的 n 位数 $l,r(l<r)$，设 $[l,r]$ 中所有由 4,7 构成的数依次为 $a_1,a_2,\ldots,a_m$，求 $a_1a_2+a_2a_3+\cdots+a_{m-1}a_m$。

$n\le 10^5$

算法

由于长度固定为 n，先考虑 $[444\cdots 4, 777\cdots 7]$ 之间的数的答案，其为 $f[n]$。则 $f[n+1]$ 可由 $f[n]$ 推出，它一定是前面加个 4，或前面加个 7，或是 $4777\cdots 7\times 7444\cdots 4$。求 $f[n+1]$ 时还需预处理 n 位的所有合法数字的和。

最终答案为 $[444\ldots 4,r]$ 的答案减去 $[444\ldots 4,l]$ 的答案，然后像数位 DP 一样做，固定一个前缀，后面可以任取。注意这时也有类似 $\mathrm{X}4777\cdots 7\times \mathrm{X}7444\cdots 4$ 的情况。

时间复杂度：$O(n)$**，空间复杂度：**$O(n)$

1.73 243D Cubes

试题大意

一个n*n的正方形地基，（i,j）上有一座高为$w_{i,j}$的塔，塔都是由1*1*1的木块搭成的。现从无穷远处向$(vx, vy, 0)$的方向看去，能看到几个木块。

$n \le 1000$

算法

用扫描线垂直于$(vx, vy, 0)$扫过去，每个格子用区间代替。问题成了：区间max，求区间min，线段树!

时间复杂度：$O(n^2 logn)$**，空间复杂度：**$O(n^2 logn)$

1.74 238E Meeting Her

试题大意

Urpal住在一个大城市。他已经计划今晚去见他爱的人。

这个城市有n个交叉点编号从1到n。交叉点被m条有向的街道连接，所有的道路有相等的长度。Urapl住在交叉点a，约会被安排在一个在交叉点b的餐馆中。他想乘坐公共交通去交叉点b。这里有k个公交车公司。在每一秒的开始，一个公交车从第i个公司随机选择一条从交叉点si到交叉点ti的最短路径然后通过这条路径。可能没有从交叉点si到交叉点ti的路径。在这种情况下没有公交车会离开si去ti。如果一个公交车通过Urpal所在的交叉点，则Urpal可以上这辆公交车。他可以在中途任意一个交叉点下车。

现在Urpal想知道是否有可能乘坐公共交通在有限的时间（旅途的时间为每条经过的边的长度和）内到达约会地点，和最坏情况下他需要乘坐公

交车的次数。

在任何时刻Urpal只知道他自己的位置和约会地点。当他上了公交车时他只知道这辆公交车属于第几个公司。当然Urpal知道城市地图和每个公司的(si,ti)。

注意Urpal不知道公交车的速度。

$n, k \leq 100$

算法

一辆公交车的行驶路线就像一张拓扑图一样。用$f(x)$表示从x点开始走最少交换几次车能到终点。然后不停的迭代更新。每次迭代可以选择一辆肯定会到x点的公交车（最坏情况下有些车根本不到x点），转移时在拓扑图上做dp就行了。设g(y)为这辆公交车从y点开始到ti，最坏情况下Urpal下车后要换车几次。有：

$g(y) = min(max(g(z)), f(y))$（拓扑图上y到z有边）

时间复杂度：$O(n^4)$，空间复杂度：$O(n^2)$

1.75 251D Two Sets

试题大意

有n个数ai，你要把他们分成2份，要求每一份xor值加起来最大，相同时要第一份xor值最小。

$n \leq 10^5$, $ai \leq 10^{18}$

算法

设总的xor为x，我们要找一个y使得x+(x xor y)最大，y越小越好。那就高斯消元，设置优先级，对于x 中是0 那位的，他可以变成1和1，使答案变大，要先做。是1那位的，无论怎么弄它都是0和1，要后做。

时间复杂度：$O(nloga_i)$**，空间复杂度：**$O(n)$

1.76 351D The Red Button

试题大意

假如你有一个n元素的数列a，你每次可以这样操作：

找v,t,k，满足$a[v] = a[v+t] = \cdots = a[v+tk]$

然后把它们全部删除，再由你来重排一遍。

现在有一个n元素的数列b，有m个询问，每次问你将$[li, ri]$的数字取出来后的数列清空最少要几步。

$n \leq 10^5, m \leq 10^5$

算法

有重排在，每次肯定可以将一种颜色的清空，除了第一次。问题成了2个：求一段数列中有多少互不相同的数字。求一段数列中是否有一种颜色的数字间隔都相同。

离线+树状数组。

时间复杂度：$O((n+m)logn)$**，空间复杂度：**$O(n)$

1.77 293E The Red Button

试题大意

你得到了一棵包含n个点的树，树上的每条边有一个非负边权，树上两点间路径的长度是该路径包含的边数，树上两点间路径的权重是指该路径包含的边的边权之和。

我们说两点是“相邻”的，当且仅当，存在一条连接该两点的路径，满足该路径的长度小于等于L,且权重小于等于W。

统计有多少个点对(u,v)，满足u ¡ v，且u,v是相邻的。

$n \le 10^5$

算法

点分治+树状数组

时间复杂度：$O(nlog^2n)$，空间复杂度：$O(n)$

1.78 348E Pilgrims

试题大意

N个点的树，有m个是城镇。一开始，每个城镇的人把离这个城镇最远的那些城镇记下来。现在你要摧毁一个城镇。如果一个城镇无法到达所有记下来的城镇，则称此城镇为不好的。最大化不好的城镇，求方案数。

$2 \le m < n \le 10^5$

算法

树形dp，分别求一个节点它的子树中与非它的子树中最远点的lca。最后每个节点可以使他不好的都是树上的一条路径。用前缀和搞一遍，选个最大的点割掉就好了。

时间复杂度：$O(n)$，空间复杂度：$O(n)$

1.79 241E Flights

试题大意

一张n个点m条边的图，你要给每条边定一个1或2的权值，使得1到n所

有路径的权值和都相同。

$n \le 1000, m \le 5000$

算法

将那些从1走走不到的和从n反着走走不到的点都去掉（与1到n的路径无关），要1到n所有路径权值和相同，其实要1到每个点都满足才符合要求。设dis[i]为1到i的最短距离，这道题就是个差分约束系统。

时间复杂度：$O(nm)$，空间复杂度：$O(n+m)$

1.80　254D Rats

试题大意

一个n*m地图，地图上有空地和障碍。空地中有一些老鼠。现要求在其中两个空地上扔两个手雷。手雷所在地用四联通的方式走d步，可以走到的空地中的老鼠会被清掉。求方案。

$n, m \le 1000, d \le 8$

算法

取其中一个老鼠，离他d的距离内一定有一个手雷，枚举一下。再取剩下的一只老鼠再在d距离内枚举一下。模拟它爆炸的过程即可。

可以用set维护剩下的老鼠。

时间复杂度：$O(d^6)$，空间复杂度：$O(nm)$

1.81 269D Maximum Waterfall

试题大意

N个平台，每个平台都是平行于Ox的线段，还有2个上下边界。平台i能向平台j流水需要：

1. $High_i > High_j$
2. $Min(R[i], R[j]) > Max(L[i], L[j])$
3. 不存在k使得i能向k流水，k能向j流水。

流的水量为$Min($之前的水量，$Min(R[i], R[j]) - Max(L[i], L[j])$ 求从上边界流到下边界的最大水量。

$n \le 10^5$

算法

高度从高向低扫描，用set维护平台组成的区间。加入一个区间时在set中枚举所有能到达它的区间，算算它最多能接收多少水。然后把这些区间删掉，因为被挡住了（两边的多余部分判一下）。

时间复杂度：$O(nlogn)$，空间复杂度：$O(n)$

1.82 346E Doodle Jump

试题大意

有n层楼，对于一个$x(0 \le x \le n)$ 会生成一个高为$ax\ mod\ p$的楼层。求楼层之间的最大间距。保证a 与p 互质。$a, x, p \le 10^9$，数据组数T $\le 10^4$

算法

以a=5,p=23为例，每次超过p都换行：

0 5 10 15 20

2 7 12 17 22

4 9 14 19

1 6 11 16 21

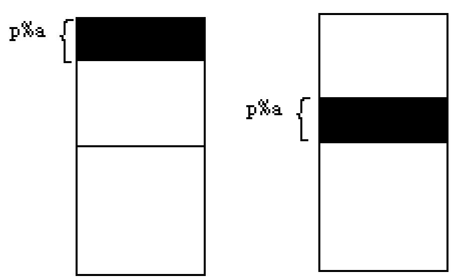
3 8 13 18

一些发现：每列中的数字都在第一排的相邻两个数字之间。间隔为p mod a。在每列的数字 mod a 下都相同：

15,17,19,16,18

0,2,4,1,3

显然最大间距更会出现在[15,20]中，而不是[0,5],[5,10],[10,15]中，因为前几排都相同，就它没有最后一排。至于[20,23]，它不会比[15,20]优的理由：



图上是楼，被分成[15,20]与[20,25]两段。阴影代表不存在的。如果我们把[20,25]倒过来也没什么关系，这样我们发现[15,20]的楼层与[20,25]的楼层完全相同，且是[20,25]先开始的。

我们发现从

15,17,19,16,18 到

0,2,4,1,3 时

a’=p mod a n’=an/p (有可能少一排要减1), p’=a

这样仍然不能保证时间复杂度度，因为现在0与p都是有楼的，翻转也没事，所以a’=min(p-p mod a, p mod a)，效率是log p

时间复杂度：$O(Tlogp)$**，空间复杂度：**$O(1)$

1.83 264D Colorful Stones

试题大意

有两串彩色的石头a与b。每块石头的颜色是红、绿、蓝之一。

初始时有两个人都站在两串石头的第一个。然后你可以发出一些指令。指令有红色、绿色、蓝色三种。在发出指令c后，站在颜色为c的石头上的人会移动到后面的一块石头上。你不能发出能使某动物走到石头串之外的指令。

两个人的位置的二元组称为一个状态。求有多少不同的可以用指令到达的状态。

长度 $n \leq 10^6$

算法

枚举第一个人走到了哪里，维护第二个人最左走到的地方与最右走到的地方。但是最左最右的区间内有一些点是走不到的。

通过一些分析，可以得出：如果第一个人走到了点i，他那串石头的第i个与第i-1个颜色相同的话，那第二个人的区间里所有点都可以走到。否则因为第一个人从i-1走到了i，发出这个指令的同时，第二个人无法待在与之同一个颜色的地方的（因为必须要走一步），但是之后可以乱发颜色不是a[i]的指令，所以第二个人到不了的点其实是b[j]=a[i-1]且b[j-1]=a[i]的点。之后还可以考虑下一步，当a[i]的指令发出后，上述的不能走的点都被走到了。所以说不能走的点只和i与i-1的颜色有关，所以上面的推理是对的。

之后只要实时维护区间的左右端点即可。

时间复杂度：$O(n)$**，空间复杂度：**$O(n)$

1.84　280D k-Maximum Subsequence Sum

试题大意

一个序列a[1..n]，两种操作。

1.修改一个数ai

2.询问一段区间内的最大k段和。

$n \le 10^5, k \le 20$

算法

有一种算法就是根据费用流做的：将a序列建成线段树，询问的时候像费用流一样取k次最大值然后点权反一下就好了。

另一种算法就是直接建线段树，维护取$0 \sim 20$段的最大值，发现合并两段区间时的dp全是单调的，复杂度同第一种算法相同。

时间复杂度：$O(nklogn)$**，空间复杂度：**$O(nlogn \;\; or \;\; nklogn)$

1.85　243E Matrix

试题大意

考虑一个大小为n*n 的，仅包含0、1的矩阵，当此矩阵满足以下条件时，此矩阵称为好的：在每一行中，所有的1都靠在一起。即，每一行都形如00...0011...1100...00（有可能是全部为0或全部为1）

给你一个n*n 的仅包含0、1的矩阵a，你的任务是判断是否能够通过重新排列某些列使得这个矩阵变成好的矩阵b。

$n \le 500$

算法

pq树裸题

时间复杂度：$O(n^2)$，空间复杂度：$O(n)$

1.86 293D Ksusha and Square

试题大意

给一个n个点的凸包，在里面随机选两个不同的格点，求这以两个点连线为对角线的正方形面积期望。

$n \leq 10^5, |xi|, |yi| \leq 10^6$

算法

根据正方形面积公式可以将x坐标与y坐标分开算。对于一个坐标，先切大条求出同一个坐标上点的个数，求$\sum_i \sum_j (x_i - x_j)^2$ 的话只要把平方拆开就能算了。

时间复杂度：$O(n + 10^6)$，空间复杂度：$O(n + 10^6)$

1.87 306C White, Black and White Again

试题大意

Polycarpus的生活总是满足“一些好事，然后一些坏事，然后一些好事”这样的规律。所以Polycarpus认为接下来的n天也是满足这样的规律的。

Polycarpus知道，接下来会发生w件两两不同的好事和b件两两不同坏事，每天至少发生一件事，每天要么全部发生好事要么全部发生坏事。

由于Polycarpus的规律，这n天会先有若干天发生好事，再有若干天发生坏事，再有若干天发生好事。(若干代指>0)

要求统计事件发生的方案数（每天发生的事的顺序也不一样），答案取模10^9+9输出

$n, w, b \leq 4000$

算法

枚举好事发生了x天，坏事发生了n-x天，算出各自的方案数，然后将所有坏事塞入好事的某一处即成了先若干天发生好事，再有若干天发生坏事，再有若干天发生好事的样子。

时间复杂度：$O(n)$，空间复杂度：$O(n)$

1.88 317C Balance

试题大意

有n个点，m条边，每个点的容量上限都是V。初始时每个点有ai的水，每次可以流一些水到相邻的点中去，但不能超过上限。求使这些点最终变成bi的水的方案。要求步数不超过$2*n^2$

$n \leq 300, V \leq 10^9, m \leq 50000$

算法

每次把一个水多出来的点流到水少的点就够了。因为每次都会使一个点完美，所以一共要n次。问题就变成用2n的步数将一个点的水运输x到另一个点。

如果运输的水量x小于等于v/2，那么直接在t到s的路径上，如果有一个点有多于x的水，那就把它流向后面。否则他得到x的水后不会溢出。在v为奇数时也适用。

那么回到原问题，如果x大于v/2，分成2次运输就可以了。

时间复杂度：$O(n^2+m)$**，空间复杂度：**$O(n^2+m)$

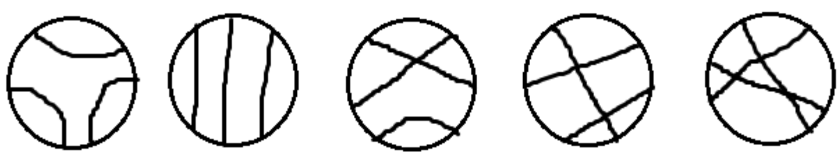
1.89 297E Mystic Carvings

试题大意

2n个分布在圆上的点，有n条路径连接它们。每个点只属于一条路径。现在要在其中选出3对点造成洞穴。一对点的距离为：从边缘走最少要经过的洞穴数+1.为了公平每对点的距离都要相同。求方案数。

算法

一共有5种可能：



要求的第一种和第五种是最麻烦的，所以可以考虑中间3种。第三种与第四种可以一起求，用线段树就行了。

时间复杂度：$O(nlogn)$**，空间复杂度：**$O(n)$

1.90 321D Ciel and Flipboard

试题大意

一个n*n的方阵，保证n是奇数。令$x=(n+1)/2$。每次操作可以将一个x*x的方阵中的数字取反。求方阵中数字和的最大值。

$n \leq 33$

算法

令$setUp[x][y]$表示(x, y)是否被取反有如下2个性质

```
1.For any i, any j<=x: setUp[i][j]^setUp[i][x]^setUp[i][j+x]
 will be 0.
2.For any i, any j<=x: setUp[j][i]^setUp[x][i]^setUp[j+x][i]
 will be 0.
It's quite easy to proof than find that: after each operation,
 there always be 0 or 2 cells lay in
 {setUp[i][j], setUp[i][x], setUp[i][j+x]} or
 {setUp[j][i], setUp[x][i], setUp[j+x][i]}.
```

只要枚举setUp[i][x](for any $i <= x$)，其他的贪心就可以了。

时间复杂度：$O(2^x * n^2)$，空间复杂度：$O(n^2)$

1.91 274C The Last Hole!

试题大意

平面上有n个点，然后每个点都以相同的速度向外扩展出一个个圆。这些圆会组成一些洞，问这些洞消失的最后时刻是多少。

$n \leq 100$

算法

暴力枚举3个点，如果他们要能组成洞，那么这3个角必须每个都小于90度，然后算出最后洞的位置，再暴力看看这个洞有没有被某个圆已经覆盖住了。

有一个问题：有4个圆组成一个洞，且4个角都是直角。（这个洞直到消失的那一刻仍然是4个圆都参与的）这个地方要特判。可以知道除了上述情况之外没有其他情况了。

时间复杂度：$O(n^4)$，空间复杂度：$O(n)$

1.92 319D Have You Ever Heard About the Word?

试题大意

若一个字符串是由某个串重复两边得到的，则称这个字符串为一个块。如：abcabc 是一个块，而abcabd与ababab不是。

现在你得到了一个串S，每次找长度最小，若长度相同最左的块，将其一半删除（重复了两次的部分只留下一个），直到不能操作为止。

问最后串的样子。

$n \leq 50000$

算法

$O(n^2)$的暴力：长度从小到大枚举，尝试着去删除。

标算：长度从小到大枚举，只有在出现这个长度的块的时候才去用$O(n)$的暴力删除。很显然这个情况下暴力只会被调用$O(\sqrt{n})$次。那么怎么判断是否出现长度固定的重复子串呢？设长度是len，我在串上面每len个分割一下。相邻的分割点求出最长的公共前缀与公共后缀。

时间复杂度：$O(n\sqrt{n} + nlog^2n)$，**空间复杂度**：$O(n)$

1.93 241D Numbers

试题大意

有一个1-n排列，按顺序选出一些数使得他们异或为0，连接起来能被p整除。

$n, p \leq 50000$，p是质数。

算法

只考虑1-31这些数，做dp。

1-31这些数有$(2^{31}) - 1$种选法，由于p是质数，所以可以假设这些选法模p域下是均匀分布的，可知模p=0 的概率非常大。

时间复杂度：$O(32^2n)$，**空间复杂度**：$O(32^2n)$

1.94 314E Sereja and Squares

试题大意

长度为n的25种括号配对。已经有一些确定的左括号。求方案数。

$n <= 10^5$

算法

F[i]表示已经有i对括号匹配的方案数。先将全部的括号看成一种，最后再乘上25的自由括号数次方。每次转移时上下界分别是$max(0, n/2 - (n - i + 1))$与$i/2$

暴力转移复杂度$O(n^2/8)$，过了。

时间复杂度：$O(n^2/8)$，空间复杂度：$O(n)$

1.95 249D Donkey and Stars

试题大意

平面直角坐标系中有n个点，你每次从一个点出发，在x-y这两条斜率的线中间再选一个点，重复这个过程。问最多有几个点。

$n \leq 10^5$

算法

变形版的最长上升子序列。把每个点按照x,y的斜率映射到y轴上，形成一个区间。一个点能到另一个点当且仅当它们的区间呈包含关系。要特判斜率为0 的情况。

时间复杂度：$O(nlogn)$，空间复杂度：$O(n)$

Chapter 2

USACO

2.1 USACO 2005 December Gold Cow Patterns

试题大意

给定一个长为n的数串A，与一个长为m的模式串B。询问A有哪些子串与B “匹配”。两个数列“匹配”是指其数字的大小关系都相同。

$n, m \leq 10^5$

算法

由于两个匹配的数列其所有对应的子序列都是匹配的，考虑直接使用kmp，这个时候kmp的匹配定义为数字大小关系都相同。对B做kmp 预处理，$next_i$ 表示最大的长度l，使得$B_{1..l}$ 与$B_{i-l+1..i}$ 的数字大小关系都相同。

现在考虑如何求出$next$数组。假如已经知道了$next_{1..i-1}$，和普通的kmp类似，用一个指针p从$next_{i-1}$开始，假如B_{1-p}中有一个数字B_j与B_p+ 1相同，则直接比较B_i与$B_{i-(p+1-j)}$的大小。如果相同的话匹配一定成功了。

若没有数字与B_{p+1}相同，那就找B_{1-p}中比它大的最小的数字与比它小的最大的数字。用这两个数字来“夹住”它。如果匹配失败了，那p只能跳到$next_p$然后继续尝试匹配。

B_{1-p}中比B_{p+1}大的最小的数字与比它小的最大的数字，这个可以使用倒推+链表快速求出。

时间复杂度：$O(n)$，空间复杂度：$O(n)$

2.2 USACO Jan 12 CowRun

试题大意

农民约翰和贝茜给奶牛们发明了一种新的运动游戏。

奶牛们在长度为$M(2 <= M <= 1000000000)$的环形轨道上从相同的位置开始跑步。这个游戏会进行$N(1 <= N <= 14)$回合，需要使用8 N张卡片,每张卡片上面都写有一个数字$X_i(0 <= X_i < M)$。每回合约翰取出最前面的8张卡片，留下前4张或者后4张，贝茜继续从约翰选出来的4张卡片中留下前2张或者后2张。接着约翰就会让奶牛们跑R*X_top的距离，R表示奶牛们已经跑过的总距离，X_top表示贝茜留下的2张中的第一张，然后贝茜会让奶牛们跑X_bottom的距离，X_bottom表示贝茜留下的2张卡片中的第二张。

约翰担心运动后，如果奶牛们跑得距离太远，奶牛就会十分疲累而不能回到出发的地方。他认为如果牛最终离他们的起始位置超过$K(0 <= K <=$向下取整(M / 2))，他们就无法回到家。我们保证如果约翰的选择(留下前4张或者后4张)总是正确的, 无论贝茜怎样选择，他总是能够确保奶牛可以回家。对于每回合,你的任务是确定约翰应该选择哪一半的卡片,来保证不管贝茜如何选择都能够使奶牛回家。贝茜将在输入中提供她的选择，而约翰的选择需要你来指定。(因为约翰并不知道贝茜会如何选择)。

算法

贝茜和约翰的游戏相当于一颗or和and的搜索树。暴搜的复杂度是$O(4^{14})$是过不了的。只要在搜索过程中加入随机儿子的搜索顺序。由于题目有解，所以根这个or节点肯定有一个儿子是true的。考虑最坏情况下的复杂度，就是一个儿子true一个儿子false的情况。有二分之一的概率会直接搜到这个儿子，另外二分之一的概率会搜进一定有一个儿子是false的and节点然后再搜另一个儿子。and节点的推导过程与or类似。设f(n) 为深度为n的子树的期望搜索复杂度，有$f(n) = 2f(n-2) + \frac{1}{2}f(n-1)$。求通项后大约是$O(1.686^{2n})$

时间复杂度：$O(1.686^{2n})$，空间复杂度：$O(n)$

2.3 USACO Mar 12 Cows in a Skyscrape

试题大意

n个物品，每个物品重ai。袋子能盛重m的物品。求最少几个袋子装完所有物品。

$n \le 18$,$ai \le m \le 10^9$

算法

f(S)是一个二元组，表示装完S这些物品，用了最少几个袋子，在袋子数相同的情况下还剩下最多多少空间。直接dp即可。

时间复杂度：$O(n2^n)$**，空间复杂度：**$O(2^n)$

2.4 USACO Dec 12 Cowstantinople

试题大意

n只牛，每只牛有一个帮派，共有m个帮派。每分钟会有一头牛进入一个牧场，如果牧场里有别的帮派的牛，那这头牛就会和其中一头牛一起离开，否则这头牛就会和自己帮派的牛在一起。求一个进入牧场的顺序，使得1 号帮派在最后占领牧场，以及最多的牛数，还要字典序最小。

$n \le 10^6$

算法

先让2-m号帮派乱搞留下尽量少的敌人，这个只要看看2-m号帮派哪个帮派牛最多就行了。然后1号帮派需要派出相应数量的牛去搞最后的敌人。先判断如果只有两个帮派，答案应该是全1，然后全2，否则1号用来占领的牛应该放在最后，1号派出去的与2-m号应该消到1个不剩。这个可以用逐位确定，在放牛之前先看看拥有最大数量的牛是否已经超过其他总数了。因为牛是按照字典序从小到大放的，所以最大数量的牛只要用后缀最大值就行了。

时间复杂度：$O(n)$**，空间复杂度：**$O(n)$

2.5 USACO Mar 13 HillWalk

试题大意

这里有n座山。每座山用一条从(x1,y1)到(x2,y2)的线段来描述，x1 <x2，y1<y2。线段不会相交或接触,包括他们的端点。此外,第一个山满足(x1,y1)=(0,0)。

奶牛贝茜从第一座山的(0,0)位置开始。当贝茜在某座山上时，她会一直爬到这座山的尽头，然后从边上跳下来。如果她落在另一座小山上,她就会继续爬那一座山;否则她就会掉到很远的地方直到她安全降落在气垫（y =-无穷大）上面。每座山(x1,y1)− >(x2,y2)视为包含点(x1,y1)但不包含点(x2,y2),所以,当贝茜掉到一个位置上满足x=x1时，就视为落到了那座山上。如果她掉到一个位置上满足x=x2时，就不能视为她落到了那座山上。

请计算出贝茜走过的山的总数。

$1 <\leq N \leq 100000$

算法

set维护扫描线上的线段。

时间复杂度：$O(nlogn)$，空间复杂度：$O(n)$