IOI2015中国国家集训队作业第一次作业

浙江省镇海中学 杜瑜皓

January 30, 2015

Contents

1 Codeforces

1.1 235C Cyclical Quest

Description

给定字符串s和n个字符串x_i，问对于每个i，有多少s的连续子串是循环同构于一个给定的字符串x_i。字符集为小写英文字母。

规定$|s| \leq 10^6, n \leq 10^5, \sum_{i=1}^{n} |x_i| \leq 10^6$。

Analysis

首先求出s后缀自动机G，并统计出每个状态在s中出现次数。对于每个t，我们要做的只是首先用G读入t，然后支持在t末尾插入一个字母，开头删去一个字母即可。在末尾插入一个字母是最基本的一个操作。开头删去字母可以沿着后缀指针往上跳，跳到长度适合即可。要注意x有可能是循环同构的，所以做串x时，每个状态最多被统计一次。

时间复杂度$O(|s| * |\Sigma| + \sum_{i=1}^{n} |x_i|)$，难度1.5/5。

Comment

这是一个两年前的题目，在当时后缀自动机没有现在那么普及，而且如果用后缀树或后缀数组都不是很好处理，如果把所有串连起来用DC3算法做出后缀数组或建出后缀树，然后在进行一系列处理也能达到线性复杂度，但是远没有后缀自动机来的简单。

1.2 235D Graph Game

Description

G是一个n个点n条边的连通图。考虑图上的分治做法，每次将图的大小计入总代价，然后在当前图中随机选择一个点删去，对剩下的各个连通子图递归处理。求总代价的期望。

规定$n \leq 3000$。

Analysis

因为期望的线性性，我们考虑总代价的期望由哪些部分构成。每个点只能被删去一次，而点u被删去时，对答案的贡献就是当前子图的大小。当前子图的大小又能被看成u被删去时所有点和u相连的概率。

令$P_{u,v}$表示u被删去时v和u相连的概率。答案就是$\sum_{u,v} P_{u,v}$。

先考虑树的情况，那么u被删去时v和u相连当且仅当u是u到v路径中第一个被删的点，所以概率为$1/d_{u,v}$。

再考虑原题，如果两个点之间没有环，那么和树上一样，不然考虑两条路径A, B，那么概率为靠A相连的概率$+$靠B相连的概率$-AB$都连通的概率，也就是$1/|A| + 1/|B| - 1/|A \cup B|$。

时间复杂度$O(n^2)$，难度2.5/5。

Comment

这是一个比较难的期望题，现场只有Petr一个人A掉了这个题。这个题从题面到解法都十分简洁自然，是我比较喜欢的一个题。

对于这个题还有一些结果，如果是树，本质上是统计不同长度的路径条数，可以通过点分治+FFT 做到$O(n\log^2 n)$的复杂度。对于本题，也可以通过类似的手法做到$O(n\log^2 n)$ 的复杂度，但是这样虽然增加了这个题的难度，但是少了很多美感。这个方法同样可以推广到仙人掌上，这样两个点的路径上可能会有很多个环，但是可以通过相同的方法，用dp统计这些环，并容斥，可以做到$O(n^4)$的复杂度。对于一般图仍然没有多项式做法。

1.3 235E Number Challenge

Description

求$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(ijk)$，$d(x)$表示$x$的因子个数。

规定$a,b,c\le 2000$。

Analysis

Solution 1

首先有一个比较自然的做法。令$dp_{p,a,b,c}$表示素因子超过p，$1\le i\le a, 1\le j\le b, 1\le k\le c$的答案，我们只要枚举$i,j,k$中因子就行了。假设下一个素数为$q$，那么$dp_{p,a,b,c}=\sum_{x,y,z}dp_{q,a/p^x,b/p^y,c/p^z}*(x+y+z+1)$，然后递归计算。

这个做法有一个优化，如果$p^2>\max(a,b,c)$，那么剩下的数至多只有一个因子，可以直接计算。但是实现起来时不方便的，需要一些琐碎的讨论。同时注意到$\lfloor a/x\rfloor$至多$O(\sqrt(x))$中取值。记$n=\max(a,b,c)$，所以这个dp的状态数是$O(n^2/\log n)$的。

考虑转移，状态(p,a,b,c)的转移代价是$O(\log_p a*\log_p b*\log_p c)$，考虑最差的情况，即当$(a,b,c)=(n,n,n)$时，当$p$为$n^{1/3}$到$n^{1/2}$ 之间的素数时转移的代价为常数，所以总代价为$O(n^{1/2}/\log n+n^{1/3}*\log^3 n)=O(n^{1/2}/\log n)$，均摊转移代价时$O(1)$ 的。

所以这样做的复杂度是$O(n^2/\log n)$。

Solution 2

这题还有一个比较有趣的做法。记

$$f(a,b,c)=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(ijk)$$

$$g(a,b,c)=\sum_{(i,j)=(j,k)=(i,k)=1}[\frac{a}{i}][\frac{b}{j}][\frac{c}{k}]$$

记$g(a,b,c)=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}h(a,b,c)$，通过容斥可得

$$h(a,b,c)=\sum_{(i,j)=(j,k)=(i,k)=1}([\frac{a}{i}]-[\frac{a-1}{i}])([\frac{b}{j}]-[\frac{b-1}{j}])([\frac{c}{k}]-[\frac{c-1}{k}])$$

为了证明$f(a,b,c)=g(a,b,c)$，只需证明$d(abc)=h(a,b,c)$即可。

注意到当且仅当$i|a$时，$[\frac{a}{i}]-[\frac{a-1}{i}]=1$，否则为0。所以$h(a,b,c)$表示三元组$(i,j,k)$的个数，$i,j,k$两两互质，且$i|a,j|b,k|c$。固定一个因子$p$，即$p^x||a,p^y||b,p^z||c$，那么$d(abc)$中关于$p$的因子为$x+y+z+1$，$h(a,b,c)$中关于$p$的因子也为$x+y+z+1((0,0,0),(c,0,0)(1\le c\le x),(0,c,0)(1\le c\le y),(0,0,c)(1\le c\le z))$，所以左右是相等的。

关于$g(a,b,c)$的计算可以先枚举i，然后枚举b,c的最大公约数进行容斥，具体答案为$\sum_{1\le i\le a,x\ge 1,(a,x)=1}\mu(x)*[\frac{a}{i}]*F(i,[\frac{b}{x}])*F(i,[\frac{c}{x}])$。

其中$F(x,y)$表示$\sum_{1\le i\le y,(i,x)=1}[\frac{y}{i}]$。

时间复杂度为$O(n^2)$。

Solution 3

还有一个比较暴力的做法，固定ij，令$S(r,k)=\sum_{i=1}^{r}d(ki)$，令$k=p^aq$，其中$(p,q)=1$。

那么通过一些简单的推导可以得到$S(r,k)=(a+1)*S(r,q)-a*S([\frac{r}{q}],q)$。

那么暴力递归即可，时间复杂度$O(n^{2.5})$，空间复杂度$O(n^{2.5})$，时间是可以接受的，空间需要一些优化。

难度2/5。

Comment

在当时比赛的时候我想到了第一个算法，但是那个时候毕竟too young，不怎么会分析这个做法的复杂度，然后就滚粗了。

比较一下这三个算法，第一个算法是比较容易想到的，通过一些优化可能达到很好的复杂度，也是大多数的人的做法。第三个做法完全就是暴力，强行枚举并固定ij，这并没有很好的利用这题的性质，所以复杂度比较差。但是要发现这个式子能进行递归计算需要一定的经验，并且其他许多常见数论函数能用相应的手段进行计算。而第二种做法利用了一个恒等式，这个恒等式显然是可以向多维的情况推广，比赛的时候只有rng_58使用这种方法通过了这个题目，但是比赛时他也不会证明，而是通过猜测得出了这个结论。我也不是很清楚怎么从正面得出这个式子，并且这种方法怎么向其他数论函数推广。算法三由Ruchiose提出。

关于这样数论函数求和，一元是平凡的，部分二元的函数往往能通过一些推导或者尝试固定其中一维得到比较优秀的复杂度，但是三元及以上往往没有比较好的手段去处理了。

类似的题目：Projecteuler 439，求$\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma(ij)$，其中$\sigma(x)$表示$x$ 的因子和，此题存在$O(n^{2/3})$的做法。

Projecteuler 432，求$\sum_{i=1}^{n}\phi(510510*i)$，此题存在$O(n^{2/3})$的做法。

DZY loves Math 4，求$\sum_{i=1}^{n}\sum_{j=1}^{m}\phi(ij)$，其中$n\le 10^5, m\le 10^9$，这题是上面那个题的简单的推广。

1.4 238D Tape Programming

Description

有一个由数字和“<”,“>”构成的程序。程序运行时有一个指针，最开始指针的指向最左字符，移动方向为向右。

我们重复以下操作直到指针指向串外：

- 如果指针指的位置是一个数字，输出这个数字并减一，然后将指针沿着原来方向移动。如果原来的数字为0则删除这个数字。
- 如果指针指的位置是“<”或“>”，那么指针的移动方向改为相应方向，并将指针沿着新的方向移动。如果新的位置也不为数字，则删除原来的字符。

现在我们有一个由程序S，你需要回答q个询问。每个询问会给你两个数l, r，如果把$S[l\cdots r]$看出一个单独的程序，问每个数字会被输出多少次。

规定$|S|, q\le 10^5$。

Analysis

首先我们可以使用链表模拟一下原程序，并且可以在源程序左边加入开始记号，即当原程序从左边结束时强制继续运行，然后维护出每个位置第一个出现的时刻，并且维护从i位置开始，下一次到i之前位置的时刻，这个可以使用树状数组解决。

容易发现，一段子程序运行过程一定是原程序运行过程的一个子段。从l开始，可以从左边或右边结束，从左边就是下一次到l 前某个位置的时刻，到右边就是到$r+1$这个时刻，取最小值即可。

时间复杂度$O(|S|\log|S|)$，难度1/5。

1.5 238E Tape Programming

Description

有一个n个点m条边的图，有k条路线从a_i到b_i，为a_i到b_i最短路中随机选择的一条。小明一开始在s上，要去t，他可以选择一条经过当前点的线路，并且在线路上某一点离开。问最坏情况下能否从s到达t，并最少要选择多少条线路。注意小明知道他当前的位置。

规定$1\le n\le 100$。

Analysis

首先一个人的状态可以有一个二元组唯一确定，即他在哪个点上，他在哪条线路上，记作(a, b)，那么可以设计状态为$dp_{a,b}$表示状态为(a, b)到终点最小的距离。

转移也很简单，在(a, b)这个点有两种决策，一种是换到另一条路线，一种是沿着路线b走下去，若换到路线c，那么c必需要一定经过点a，这个可以通过简单的预处理得到。那么答案为$\min(dp_{a,c}+1|c$为可换乘路线)。沿路线b走下去，那么答案为$\max(dp_{c,b}|c$ 为从a沿路线b 能走到的下一个点)。

但是这样的转移是没有拓扑关系的，我们可以通过迭代，把能到达t点的状态不断加入，知道没有状态加入即可。

每个状态至多更新$O(n)$次，每次更新的代价是$O(n)$。

时间复杂度$O(n^4)$，难度1/5。

1.6 240F Torcoder

Description

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字典序最小的回文串。输出经过m次操作后的字符串。

规定$n, m \leq 10^5$，字符集为小写字母。

Analysis

建26棵线段树，维护区间内字符出现次数，重排只需统计每个字母出现次数，然后将其中一半从a到z排序即可，如果有一个字母出现了奇数次，那么放在中间。这就是一个简单的区间赋值操作。

时间复杂度$O(m \log n * |\Sigma|)$，难度0.5/5。

1.7 241B Friend

Description

给定n个数$a_1, a_2, \cdots, a_n$，求两两异或值前k大的和。

规定$n \leq 50000, a_i \leq 10^9$。

Analysis

首先把这些数建成一棵Trie。

然后二分答案x，统计多少对数异或和不小于x，即对每个数a_i，统计哪些数异或起来不小于x即可，这个只要在Trie上遍历一下即可。

然后对于每个数a_i，统计异或起来超过x的异或值的和，这个我们可以分别考虑每一位，所以只要知道那些数字某一位0或1的个数即可。只要在Trie上多记录一点信息即可。

时间复杂度$O(n \log^2 W)$，其中W为a_i的范围，难度0.5/5。

Comment

这个题Codeforces上时限是$6s$，tsinsen上时限是$0.2s$。

1.8 241D Numbers

Description

给定一个$1, 2, \cdots, n$的排列a，删除一些整数,使得结果序列满足以下三个条件：

- 新序列非空。
- 序列中所有数异或和等于0。
- 把所有数按十进制从前往后写在一行当成十进制数，这个数被p整除。

规定$n, p \leq 50000$,，且p为质数。

Analysis

我们只保留不超过31的数字，然后暴力dp，记录当前序列异或和，模p的余数。

时间复杂度$O(p\log^2 p)$。难度2/5。

Comment

其实我无法证明这个做法的正确性，我甚至给不出一个上界。我们来直观上理解一下，$1, 2, \cdots, 31$中异或和为0的非空子集大概有$2^{26} - 1$个。由于p的质数，我们把它当成一个随机函数，那么无解的概率是很小的。所以这里的只要保留$O(\log p)$ 个数字即可。

并且，由于题目给出的几个条件，几乎没有联系，并且不能进行进一步分析(如果数字连起来要补0补齐，或许还能从10^x模p的循环节出发进行进一步分析)，也在某种程度上暗示我们要进行一些比较大胆的做法。

由于没有官方题解，我也并不知道出题人到底怎么想的。

1.9 241E Flight

Description

有一个n个点的有向无环图，要求把每条边的权值改为1或2，使得从1到n到所有路径长度相同。

规定$n \leq 1000, m \leq 5000$。

Analysis

容易发现，所有路径长度相同意味着每个能在1到n的路径上的点有一个编号L_i，使得$w_{u,v} = L_v - L_u$，又因为$1 \leq w_{u,v} \leq 2$，所以得到限制$1 \leq L_v - L_u \leq 2$，这是个差分约束问题，所以存在整数解。

时间复杂度$O(nm)$，难度0/5。

Comment

一开始我不会做这个题，因为我误以为这样不一定能得到整数解，然后想了一些迭代之类的做法，以为这是个很难的题。

1.10 241F Race

Description

阅题理解。

Analysis

模拟，难度0/5。

Comment

真noip普及组。

1.11 243C Colorado Potato Beetle

Description

有一个$(10^{10}+1)*(10^{10}+1)$的矩形，一开始小明在中间那个点上，他会在上下左右四个方向中选择一个并移动整数米，并且把他经过的格子设置为不可经过，他会移动n次。问有多少格子是从边界不可达的。

规定$n \leq 1000$。

Analysis

离散化，bfs。

时间复杂度$O(n^2)$，难度0.5/5。

1.12 243D Cubes

Description

有一个底座$n*n$的积木，问向量$v=(v_x, v_y, 0)$的平行光能照到多少立方体。我们认为这个立方体是可见的，当且仅当立方体上存在一个点p，从p以向量$-v$发出射线上不存在属于其他立方体的点。

规定$n \leq 1000$。

Analysis

首先一个直观的认识就是每堆积木覆盖光线的一个区间，并且有先后关系，如果一个积木对应的区间内的每个位置都存在某个积木在它前面，那么它就不能被照到。

具体一堆积木对应的区间就是要找到四个顶点中最左边和最右边，至于如何定量判断左右，可以在每个顶点沿着向量做直线，然后投影到与向量不平行的直线上即可，也就是顶点与向量的叉积。用其中一个顶点与向量的点积当成这堆积木的深度。虽然一堆积木的深度不能用其中一个点简单地代替，但是这么做能保证如果两堆积木对应左右区间如果有交，那么用这样做是没问题的。

接下来就按深度从前往后，依次处理每个积木。记a_i表示第i个区间当前积木高度，只要在区间$[l, r]$找到一个最低的位置a_k，就能计算出当前这堆积木能被看到$\max(h-a_k, 0)$个，然后用这些积木更新这段区间，也就是将区间$[l, r]$之内的数a_i变成$\max(a_i, h)$。

这个标记可以合并，所以可以使用线段树完成。

时间复杂度$O(n^2 \log n)$，难度1/5。

1.13 248E Piglet's Birthday

Description

又n个可重集，每个集合有若干个数，起初每个数都是1。有q次操作，从集合u_i随机拿走k个数字，全都改为0，加入集合v_i中。请问每次操作后全是0的集合期望个数。

规定$n, q \leq 10^5, a_i \leq 100, k \leq 5$。

Analysis

首先注意两个事实，每个集合中1的个数不增，并且每次操作之后集合总个数是确定的。

于是令$dp_{i,j}$表示集合i中有j个1的概率。答案就是$\sum_{i=1}^{n} dp_{i,0}$。

对于每次修改操作，只改变dp_{u_i}，只要枚举k个数中有几个是1，然后计算概率，进行转移即可。

时间复杂度$O(qkW)$，其中$W = \max(a_i)$，难度0.5/5。

1.14 249D Donkey and Stars

Description

在n个点的点集中选择若干个点，使得每个点在前一个点的右上方，且连线与O_x夹角在$[\alpha_1, \alpha_2]$内。规定第一个点的前一个点为原点。求最多能取多少个点。

规定$n \leq 10^5$。

Analysis

把坐标轴拉伸一下容易发现这就是最长不降子序列。

时间复杂度$O(n \log n)$，难度0/5。

1.15 249E Endless Matrix

Description

给定一个无限大的矩阵。形如下：

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

有q个询问，求一个子矩阵$w_{i,j}(a \le i \le b, c \le j \le d)$的和，如果答案不小于$10^{10}$，那么输出$\cdots$和答案的后十位，否则输出答案。

规定$q \le 10^5, a, b, c, d \le 10^9$

Analysis

首先一个子矩阵显然可以通过容斥转化成四个顶点在$(1,1)$的矩阵的线性组合。

假设这个矩形的顶点是$(1,1),(w,h)$。不妨设$w < h$，那么可以划分成一个正方形$(1,1),(w,w)$ 和一个矩形$(1,w+1),(w,h)$。

正方形里的数是$1 \sim w*w$，然后矩形每一列的和构成一个等差数列，首项和公差简单的求和即可。$w >= h$时同理。

然后题目要求判断答案是否小于10^{10}，有几种解决方法。

- 高精度。
- 对10^{10}取模，同时使用long double估计答案。
- 取接近10^{10}多个模数，然后看答案取模是否相同。

时间复杂度$O(q)$，难度0/5。

Comment

这题我已经在另一篇题解吐槽过了。然而前面连续三个题都是同一个人的一场div 1里面的CDE 题，我觉得难一点的div 2都比这个难了。

1.16 251D Two Sets

Description

有一个n个非负整数的集合$A = \{a_1, a_2, \cdots, a_n\}$，划分成两个集合，即$x_1$为第一个集合的异或和，$x_2$为第二个集合的异或和。求一种划分方案，在$x_1 + x_2$最大的情况下使得$x_1$最小。

规定$n \le 10^5, a_i \le 10^{18}$。

Analysis

首先可以发现$x_1 \oplus x_2$是已知的(这里的$\oplus$符号表示异或)，即为所有元素的异或和，记作s。那么如果确定了x_1，x_2也是确定了。

考虑一位一位的确定x_1，由于要求$x_1 + x_2$最大，如果x的第i位为0，那么要求x_1 的第i位为1，记为第一类要求。如果x的第i位为0，那么x_1的第i位为0或1都不影响答案，但由于x_1要尽量小，那么要求x_1的第i位为0，记为第二类要求。

那么由于要求$x_1 + x_2$最大，所以第一类要求的优先级高于第二类要求的优先级，同类要求之间位数越高优先级越高。

我们要做的就是讲要求按优先级从高到低，如果当前要求在当前限制下能满足，那么将这个要求加入限制，否则跳过。

这是一个经典的问题，只要将所有数字的每一位按优先级重新排列，然后使用高斯消元求出这些数的基，然后贪心即可。

时间复杂度$O(n \log W)$，其中W为a_i的上界，难度1/5。

1.17 251E Tree and Table

Description

求把一棵$2n$个结点的树放入2行n列的矩形表格的方案数，满足：

- 格子和结点一一对应。
- 如果两个树上的节点间有一条边相连，那么它们放入的格子有一条公共边。

规定$n \le 10^5$，答案对10^9+7取模。

Analysis

首先我们注意到每个点度数不超过3，因为每个格子相邻的格子不超过三个。

如果每个点的度数不超过2，也就是一条链，显然可以通过手算得出答案。如果链的两端固定，那么放入的方式是固定的，证明几乎显然。那么只要考虑两端的可行方案即可，首先两端的奇偶性不同，如果不在第1列或第n列，那么另一个端点不能放在同一列中，如果$n \ne 1$，那么答案就是$2(n-2)(n-1)+4n = 2n^2-2n+4$，如果$n=1$，答案为2。

假设现在图中有一个度数为3的结点u，枚举哪两个点放在这个点的左右两边，那个点放在对边，记对边点为v，左右点为l,r。令dp_x表示以u 为根x这棵子树恰好放在2行的表格中且x为表格边上的一点的方案数，恰好表示放满整个表格。令$f(x,y)$表示以u为根x,y这两棵子树恰好放在2行表格中，且x,y分别为表格同一边上两点的方案数。

如果v的度数为1，那么答案就是dp_l*dp_r，如果v的度数为3，那么记与v相连的点为u,v_l,v_r，那么答案为$f(v_l,l)*f(v_r,r)+f(v_l,r)*f(v_r,l)$，即$v_l$与$l$放同一边，或者放不同一边的方案数。如果$v$的度数为2，记与$v$相连的点为$w$，那么答案为$f(w,l)*dp_r+f(w,r)*dp_l$。

先考虑如何求dp_x，首先如果子树x的大小为奇数那么答案一定为0。

如果x的度数为3，那么除了父节点之外两个结点分别为l,r，考虑l与x放在同一列之内。如果l度数为3那么显然是无解的。如果l度数为1，那么答案为dp_r，如果l度数为2。记与l连的另一个点为w，那么答案为$f(w,r)$。r与x在同一列同理。

如果x度数为2，那么沿着儿子往下走找到第一个度数不为2的点v。由奇偶性，v的在某一列中的上下位置是确定的，所以不必考虑。

如果v度数为1，也就是整个子树是一条链，那么摆放的形态由另一端点确定，由于奇偶性不同，所以方案恰好为子树大小的一半。

如果v度数为3，记v 父节点为f，另外两个结点为l,r。默认x在的一端为右边。

如果v的同列的是f，如果r在v的右边，那么r的子树必须是一条链，且长度小于x到f链的长度，答案就是dp_l，l在v的右边同理。

如果v的同列是l，那么f在v的右边，接着讨论l的度数。

如果l度数为1，那么答案显然是dp_r。

如果l度数为3，另两个点记为a,b，如果a放在右边，那么一定有a的子树是一条链，且长度不超过x到f链的长度，于是答案为$f(b,r)$，b放在右边同理。

如果l度数为2，记另外一个点为a，如果a放在右边，限制同上，答案为dp_r，a放在左边答案为$f(a,l)$。

接着考虑如何求$f(x,y)$，也就是把两棵子树放满2行格子的方案，如果两棵子树都不是链那么一定无解的，不妨假设x为链，那么以y开始的链如果小于x开始链的长度，那么也是无解的，不然从y开始去掉长度与x链相同的链，记此时结点为w，那么答案即为dp_w。两条链且长度相同需要特判一下。

实现时要求每个点一下链的长度，以及某条链上的某个点为什么，这个只要记录每个点以下第一个度数不为2的点，以及这个点上面对应的整条链就行了。

最后由于上下对称和左右对称，需要将答案乘上4。

时间复杂度$O(n)$，难度3/5。

Comment

这个题应该是我做的Codeforces部分里面最难的一个题，当然有其他更难的题目不过我没有做。虽然这个题只要不停地讨论就行了，但是我还是挺喜欢的，因为题目本身是个很简洁很自然的问题。

我做的时候漏讨论了一种情况，这挺不应该的，不过我觉得在两小时的比赛中做出这个题的可能性真的不是很大。

1.18 253E Printer

Description

有一个从0时刻开始工作的打印机，每秒中打印一张纸。

有n个任务，每个任务有三个参数t_i,s_i,p_i，分别表示打印机接收到任务的时间，打印的张数，及优先级。所有任务优先级互不相同。

打印机的工作方式是每秒钟在接收到但未完成的任务中选择优先级最高的任务。

然而有一个任务你知道除了优先级以外的信息，并还知道完成的时刻T，求出这个未知的优先级，并求出每个任务完成时刻。

规定$n \le 50000, t_i, s_i, p_i \le 10^9, T \le 10^{15}$。

Analysis

Solution 1

对于一个任务，随着优先级的下降完成的时间不会往前，所以我们可以二分优先级，然后使用优先队列维护打印机的状态即可。

时间复杂度$O(n\log n\log T)$，难度0.5/5。

Solution 2

还有一个有趣的做法。

首先把未知的任务i优先级设为最低，进行一遍模拟。容易发现这样一个事实，就是任务i的完成时间肯定在T之后，并且在t_i到T这段时间打印机肯定是不空闲的。

如果不断提高任务i的优先级，那么这一段时间内优先级低于i的任务将会被延迟到时间T后执行。所以将t_i到T这段时间内的任务按优先级排序，不断提高任务i的优先级直到可以时间T完成即可。

时间复杂度$O(n\log n)$，难度1/5。

1.19 254D Rats

Description

有一个$n*m$的网格，有些格子是被墙占据，有些格子有老鼠，其余格子是空的。

小明要在网格中放两个手榴弹，一个手榴弹能覆盖离这个这个格子最短路不超过d的格子，问是否有存在可行方案，使得两个手榴弹覆盖所有的老鼠。

规定$n,m\le 1000,d\le 8$。

Analysis

首先注意到一个手榴弹能覆盖的格子个数是$2d^2+2d+1$的，所以老鼠个数至多是$4d^2+4d+2$。

枚举每个老鼠，然后再枚举每个能到这个格子的位置，求出这些格子能覆盖到哪些老鼠。然后再枚举选哪两个格子。

可以使用bitset优化，以及对覆盖老鼠集合一样的进行判重。

时间复杂度$O(d^8)$，难度0.5/5。

1.20 256D Liars and Serge

Description

有n个人，每个人说真话或假话。

对所有人有一个问题，多少人是说真话的。说真话的人会说出正确答案，而说假话的人回答1到n之间除了正确答案之外任意一个数。

每个人的回答形成了序列$a_1,a_2,\cdots,a_n$，问有多少种序列，使得能确定恰有k个人显然说谎了。

规定$k\le n\le 2^8$，且n恰为2的幂次。

Analysis

如果一个人的回答是x，但是没有x个人回答x，那么这些人显然说谎了。

令$dp_{i,j,k}$表示考虑了答案是1到$i-1$这些人，然后已经有j个人，其中k个显然说谎了。

枚举回答了答案i的人数l，对应的方案数是$\binom{n-m}{l}$，然后进行转移。

时间复杂度$O(n^4)$，难度0/5。

Comment

因为n是2的幂次，所以只要将所有答案预处理打表就行了。

这么奇葩的题当然是sereja出的。

1.21 257E Greedy Elevator

Description

有m层的楼房，其中有一台电梯。有n个人从时刻t_i来到s_i层想要到达f_i层。

电梯是这样工作的:

- 如果没有人在电梯且没有人等待，电梯不动。
- 那么若当前时刻在电梯所在楼层以上等待的人和电梯中想要上去的人不少于对应的向下的，电梯向上一层，否则电梯向下一层。

问每个人什么时候到达他想去的楼层。

规定$n, m \le 10^5, t_i \le 10^9$。

Analysis

当且仅当有人在等待，有人上电梯，有人下电梯时电梯运动状态改变，所以只要用优先队列维护这些时间点，模拟即可。

时间复杂度$O(n \log n)$，难度0.5/5。

1.22 258D Little Elephant and Broken Sorting

Description

有一个排列$p_1, p_2, \cdots, p_n$，有m次操作，第i次操作交换第a_i个数和第b_i个数，但有一半概率不执行。求经过m次操作之后逆序对的期望。

规定$n, m \le 1000$。

Analysis

令$dp_{i,j}$表示$p_i > p_j$的概率，那么答案就是$\sum_{i=1}^{n}\sum_{j=i+1}^{n} dp_{i,j}$。

对于第k次修改操作，只需改变一下$dp_{i,j}$其中i或$j \in \{a_k, b_k\}$的值即可。

复杂度$O(nm)$，难度0/5。

Comment

两年前我打过这场比赛，打出了至今为止Codeforces div 1的最差成绩。当然这场比赛还是有很多人AK的，因为这场比赛的确很水。

当时以为求$dp_{i,j}$表示第i个数为j的概率，然后就可以计算了，但是$p_i = j$和$p'_i = j'$这两个事件是不独立的，所以$p_i < p_j$的概率不能直接这样计算。当年的我还是too young。

1.23 260E Dividing Kingdom

Description

有n个点，被2条与x轴平行，2条与y轴平行的不同直线划分成9部分。

现在知道了每块点的个数，但是不知道对应关系，请找到这4条直线。

规定$n \le 10^5$。

Analysis

枚举块与点数的对应关系，然后用任意能支持二维数点的数据结构统计判断即可。

时间复杂度$O(n \log n)$，难度0.5/5。

1.24 261D Maxim and Increasing Subsequence

Description

定义数列a:

$$a_i = \begin{cases} b_i & 0 < i \le n \\ a_{i-n} & n < i \le n * t \end{cases}$$

求a的最长上升子序列。

令$m = \max(b_i | 1 \le i \le n)$。

规定$n, m \le 10^5, t \le 10^9, n * m \le 2 * 10^7$。

Analysis

将b离散化，令$m = \max(b_i | 1 \le i \le n)$，即为不同数字的个数。

首先因为要求数字递增，所以答案不会超过不同数字的个数m。如果$t > m$，那么答案显然为m。

现在假设序列不超过$n * m \le 2 * 10^7$，那么令dp_i表示以不超过i的数结尾最长上升子序列。

每次加入一个数c，它结尾的答案就是$dp_{c-1} + 1$，然后从c到m暴力更新，因为dp单调不降，所以一旦不能更新就能退出。

这样做保证dp_i最多被更新$m \le n$次。

时间复杂度$O(nm)$，难度0.5/5。

Comment

这个题的处理手法比较诡异，这种使用暴力更新，然后用均摊的手法保证复杂度，不是很多见。

1.25 261E Maxim and Calculator

Description

有一个二元组$(1,0)$，每次操作可以把(a,b)变成$(a,b+1)$或$(a*b,b)$。问有多少个正整数$x \in [l,r]$满足，在不超过p次操作之后使得二元组第一个数为x。

规定$l \le r \le 10^9, p \le 100$。

Analysis

令$dp_{a,b}$表示到达a,b状态最小的步数，然后使用BFS或dp即可。

难度0/5。

Comment

关于时间复杂度和实现细节的一些说明。首先我们发现由于步数不超过k，那么a中的素因子都不超过k，$1 \sim 10^9$之间的数字素因子不超过100的大概只有$3*10^6$个左右。可以预先求出这些数字并排序，然后转移的时候按b相同分层，层间从a转移到$a*b$，可以使用两个指针维护，那么可以比较轻松的做到线性。关于渐进时间复杂度应该是$O(\pi(k)\log^{\pi(k)} n)$，但在这个题目中远远达不到这个上界。

sereja出的div 1E往往简单得令人难以置信。

1.26 263E Rhombus

Description

有一个$n*m$的表格$a_{i,j}$，和一个非负整数k。

记$f(x,y)=\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}\max(0,k-|i-x|-|j-y|)$，求$(a,b)$满足$k \le a \le n-k-1, k \le b \le m-k-1$使得$f(a,b)$最大。

Analysis

发现$f(x,y)$表示中心(x,y)的菱形区域加权的和。考虑计算$g(x,y) = \sum_{dx\ge0,dy\ge0,dx+dy\le k} a_{x+dx,y+dy}(k-dx-dy)$，另外几个方向同理。

令$p(x,y)=\sum_{0\le dy\le k} a_{x,y+dy}(k-dy), q(x,y)=\sum_{dx\ge0,dy\ge0}^{dx+dy\le k} a_{x+dx,y+dy}$，这两个都能通过简单的递推得到。

而通过简单的计算可以发现$g(x,y)-g(x,y+1)=p(x,y)-q(x+1,y)$，那么$g(x,y)$也能通过递推得到。

时间复杂度$O(n^2)$，难度0.5/5。

Comment

即使这样，这还是一个十分琐碎的题。

1.27 266D BerDonalds

Description

给定一个n个点m条边的无向带权联通图。

求一个点u使得$\max_{v\in V}(d(u,v))$最小。

规定$n \le 200$，没有重边自环。

Analysis

首先假设u在边(x, y)上，那么$d(u, v) = \min(d(u, x) + d(x, v), d(u, y) + d(y, v))$，其中$d(u, x) + d(u, y) = d(x, y)$。

那么考虑将$d(u, v)$看做$d(u, x)$的一个函数，对于每个结点v对应图像一条折线。

所以只要按左端点排序，然后用栈维护哪条折线在最上面即可。

时间复杂度$O(n^3 \log n)$，难度0.5/5。

Comment

这个题与最小直径生成树等价，见Marc Bui, Franck Butelley, Christian Lavaulty, *A Distributed Algorithm for the Minimum Diameter Spanning Tree Problem*。早年出现于Amber的Play with Trees比赛中的C题，同时也见于唐文斌2012年冬令营讲课。

1.28 266E More Queries to Array...

Description

你有一个包含n个整数$a_1, a_2, \cdots, a_n$的数组，执行m个操作，每个操作为以下两种之一：

- 将区间$[l, r]$中的元素赋值为x。
- 计算$\sum_{i=l}^{r} a_i * (i - l + 1)^k$。

规定$n, m \le 10^5, k \le 5$。

Analysis

由于二项式定理$a_i * (i - l + 1)^k = \sum_{j=0}^{k} a_i * i^j * \binom{k}{j} * (1 - l)^{k-j}$。所以只要维护$\sum_{i=l}^{r} a_i * i^j (0 \le j \le k)$即可。

这个可以用线段树轻松解决。对于赋值操作可以对每个结点预处理$\sum_{i=l}^{r} i^j$即可。

时间复杂度$O(nk \log n)$，难度0.5/5。

1.29 267C Berland Traffic

Description

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$c_{x,y}$，那么有$|t| \le c_{x,y}$，网络中1 是源点，n是汇点，除了源汇点之外所有点流入的流量等于流出的流量。

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。

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。

规定$n \le 100, m \le 5000$。

Analysis

因为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，所以可以给每个点一个编号L_i，那么边(x, y)的流量即为$L_y - L_x$。

不妨设$L_1 = 0$，那么L_n即为网络流量。这个网络可以类比成电阻为1的电阻网络，设n号点电势为1，通过高斯消元解出每条边的电流。

那么每条边的实际流量为$|I_{u,v} * L_n| \le c_{x,y}$，可得$L_n$的范围。

时间复杂度$O(n^3)$，难度0.5/5。

Comment

这个题目要注意图不连通，$I_{u,v} = 0$等情况。

1.30 269D Maximum Waterfall

Description

有若干条线段，其中最上面线段为从$(-10^9, t)$到$(10^9, t)$的线段，最下面线段为从$(-10^9, 0)$到$(10^9, 0)$的线段，中间有若干线段，第i条为(l_i, h_i)到(r_i, h_i)的线段。

规定第i条线段可达第j条线段，当且仅当满足下面条件：

- $\max(l_i, l_j) < \min(r_i, r_j)$
- $h_j < h_i$
- 不存在k使得(i, k)与(k, j)均满足以上两个条件。

线段i到线段j的流量为$\min(r_i, r_j)$-$\max(l_i, l_j)$。

一个从最上面的线段到最下面线段的线段序列，序列中前一条线段可达后一条线段，整个序列的流量为所有相邻线段流量最小值。

求一个线段序列使得流量最大。

规定$n \le 10^5$。

Analysis

首先使用扫描线，用平衡树维护这些线段，考虑如何建这个图。

如果i可到达j，假设i先被加入平衡树，那么当j被加入时，i必定是j上面的第一条线段，否则与条件3矛盾。j先被加入同理。

所以只要在每一条线段加入时，往上面和下面的第一条线段连边即可。

当然也有例外，如果k比i和j后加入，k在i和j之间，但i和j之间已经连边，那么只要将i和j之间的边删去即可。这样做能保证扫描线当前的线段建立的图是正确的，这个只要归纳一下不难证明。

然后在DAG上dp即可。

时间复杂度$O(n \log n)$，难度0.5/5。

Comment

和WC2012记忆中的水杉树第二问差不多。

1.31 269E String Theory

Description

有一个$n*m$的网格，网格左右边界各有n个钉子，上下边界各有m个钉子。同时，有$n+m$条线段连接了若干对不同侧边界的钉子，不存在两条线段连接同一个钉子。

现在有两种操作，选择不同的两行或两列，对应地交换处于同侧的钉子，但不改变钉子与线段的连接情况。

给定初始情况，能否通过若干次操作使得线段两两不交叉。

规定$n,m \le 10^5$。

Analysis

我们只要注意到两个事实。

首先统计多少条边连接了上左，上右，下左，下右，上下，左右，那么最后的形状是唯一的。

然后如果把相对的钉子连起来，那么这个图就变成了若干个环，那么在操作过程中，并不改变这些环的构成。并且环构成相同的两个网格，可以是看成一个函数将其中的一个网格中行列映射到另一个网格中的行列，行和列分别是一个置换，一定能通过交换行列得到。

所以只要判断当前网格的环构成和目标网格的环构成是否相同，这个可以使用最小表示法解决。

时间复杂度$O(n\log n)$，难度1/5。

Comment

无论作者的预期还是这个题赛后的AC人数，好像显示着这个题很难的样子。然而我完全没有这种感觉。不过这个题还是挺不错的，大方美观干净。

1.32 273D Dima and Figure

Description

有一个$n*m$的网格，求如下子集数目：

- 集合非空。
- 集合中的格子形成一个连通块。
- 集合中格子(x_1,y_1)移动到格子(x_2,y_2)的最短路等于$|x_1-x_2|+|y_1-y_2|$，其中一个格子只能移动到四联通的且在集合中的格子里。

规定$n,m \le 150$，答案对10^9+7取模。

Analysis

第三个条件令$x_1=x_2=x$，那么条件等价于(x,y_1)到(x,y_2)的路径上个字都属于这个集合，也就是对于每个x坐标，在集合中的格子为空，为一个区间内所有的格子。y坐标同理。

那么可以发现格子的上边界先不降后不增，下边界先不增后不降。那么令$dp_{i,j,k}$表示做到第i列，边界为$[j,k]$，然后再用两维记录当前边界属于不增的部分还是不降的部分。在转移时，使用前缀和优化即可。

时间复杂度$O(n^2m)$，难度0.5/5。

Comment

这个题早年见于SGU 167，十多年前的题目了，这是一个求极值的题目，即选恰好大小为k的子集使权值和最大。

同时这个题目又见于SRM 493,现场只有Petr一个人过掉，这个题要求是有些格子不能选取的，然而做法和本题基本相同。

1.33 273E Dima and Game

Description

有一个游戏，开始有n对整数$1 \le l_i < r_i \le p$，玩家轮流操作，每次选择一对l_i, r_i 其中$r_i - l_i > 2$，将其替换成$(l_i + [\frac{r_i - l_i}{3}], l_i + 2 * [\frac{r_i - l_i}{3}])$或者$(l_i, r_i - [\frac{r_i - l_i}{3}])$。不能进行操作玩家则输。

求存在多少个初始状态使得先手必胜。

规定$n \le 1000, p \le 10^9$，答案对$10^9 + 7$取模。

Analysis

首先分析这个游戏，由SG定理得，我们只要求出每个游戏的SG值就能知道先手是否必胜。

然后我们注意到一个事实，那就是每个游戏只与$r_i - l_i$有关。记$f(n)$表示当$r - l = n$是游戏的SG值，那么$f(n) = \text{mex}(f([\frac{n}{3}]), f(n - [\frac{n}{3}]))$。显然$f(n) \in 0, 1, 2$。

可以发现$f(n)$是一个分段函数，每段取一个相同的值。我们可以预处理出$f(n)$函数，然后统计$1 \le l_i < r_i \le p$中有多少个游戏的SG值为$0, 1, 2$。

接下来就是一个简单的dp,$dp_{i,j}$表示选了i个游戏，当前游戏SG值的异或和。

时间复杂度$O(n)$，难度0.5/5。

Comment

关于$f(n)$是一个分段函数的解释。假设当前这一段从n开始，即$f(n) \ne f(n-1)$。那么就会导致$3n$和$[\frac{3n-1}{2}]$处SG值发生变化。假设段数是$g(p)$，那么可以用上述的方法在$O(g(p) \log g(p))$的代价内算出。

通过观察$f(n)$的段数是$\Theta(\log p)$的，可以通过找规律之后用数学归纳法证明就行了，证明是平凡的。

很多时候SG值表现出来奇特的性状是不容易证明的，所以在算法竞赛中只要找规律就行了。

sereja的题，特别是计数，总是令我感觉莫名其妙，往往是很多水题套了个计数的壳，变成一个拼题，让人感觉无聊而厌恶。

1.34 274C The Last Hole!

Description

有n个圆，第i个圆的圆心在(x_i, y_i)。最开始所有圆半径都为0，然后所有圆同时开始变大，在时刻$t(t > 0)$所有圆的半径都为t。我们可以想象成一些黑色的实心圆放在一个无穷大的白色平面上，每个时刻都会存在一些黑色和白色的联通快。定义一个白色的封闭区域为一个“洞”。

求最后一个洞消失的时刻。

规定$n \le 100$。

Analysis

为了更方便的研究这个问题，我们考虑这些点形成的Voronoi图。那么最后的洞一定是V图中的某个顶点，所以只要考虑一个顶点是否会形成一个洞就行了。

对于一个V图顶点，它周围若干个圆心在一个半径为r的圆上，将圆周分成若干段弧。如果一段弧的圆心角小于π，那么在时刻r之前这两个圆会在这个区域相交，也就是这个区域是封闭的。如果有一段弧的圆心角不小于π，也就是所有的点都在某条直径的同一边，那么在时刻r之前所有圆的交点还是某条直径同一边，不会产生洞。所以产生洞的充要条件就是每段弧对应的圆心角都小于π。

这样做建出V图后是线性的，但是当所有点都在一个大圆周附近，就会产生严重的精度误差，于是需要写有理数类，比较麻烦。

可以简化这个过程，可以证明如果满足条件的洞，一定是一个锐角三角形的外心，或者是一个矩形的中心。考虑一个V图的顶点，如果圆周上每个点所在直径对面都有一个点，那么两条直径对应四个圆周上的点构成了一个矩形。否则找一个点u，使得所在直径对面没有点，记u的对径点为u'，只要在圆周上找到u'顺时针逆时针的两个点v,w，因为任意一段圆弧对应的圆心角小于π，所以u, v, w构成了一个锐角三角形。

于是就可以枚举锐角三角形和矩形，然后找到对应点，判断是否是V图的顶点，即到这些点的距离是否不大于到其他点的距离，这样做并不要判断两个数是否相等，所以精度是可以接受的。

时间复杂度$O(n^4)$，难度1/5。

Comment

这个题的题解是直接证明了这个结论，但是我觉得如果不显式地建出V图，做这个题还是挺困难的。

1.35 274E Mirror Room

Description

有一个$n * m$的网格。网格中有k个堵塞的格子，其他的格子是空的。你在某个空格子的中心向一个对角线方向发射一束激光。如果光束碰到堵

塞的格子或网格边界，它会反射。过了一会儿，光束进入了一个无限的循环。

求至少被光束通过一次的空格子数。我们认为光束通过了一个格子的中心才算是通过了这个格子。

规定$n, m, k \le 10^5$。

Analysis

把边界当成格子，那么光线一定能由(a, dir)表示，代表从a格子沿dir方向反射，所以光线不超过$O(n+m+k)$段，直接模拟，然后用相关数据结构维护每条斜线即可。

最后将所有的光线合并，由于奇偶性相同，所以斜率为1的光线和-1的光线不会相交于同一格中间。所以只需分开考虑即可。

时间复杂度$O(n \log n)$，难度0.5/5。

1.36 277D Google Code Jam

Description

有n个题目，每个题目分为简单和困难两个部分。对于每个题目做出简单部分之前是不允许做困难部分的。同时对于题目i，简单和困难部分都有相应的做题时间st_i, lt_i和通过后的分数ss_i, ls_i。保证简单部分一定能通过，但是困难部分有p_i的概率不能通过。

比赛总共有t时间，请求出一个解题顺序使得得到的期望分数尽可能高，同时保证期望罚时尽可能少。罚时指最后一次通过的提交的时间。

规定$n \le 1000, t \le 1560$。

Analysis

假设我们确定了做的题目，那么期望得分就固定了，考虑怎样的顺序让期望得分尽量小。

首先毫无疑问先做完简单的部分，如果简单在t时间做完那么罚时一定为t，否则罚时有可能小于t，所以期望更小。

对于题目i和j，如果i恰在j前面，考虑这两个题顺序对期望罚时影响，显然可以仅考虑这两个题，i和j都过或都没过的情况是相同的不用考虑，那么一个罚时为$(1-p_i)*p_j*lt_i+p_i*(1-p_j)*(lt_i+lt_j)$，而另一个罚时为$(1-p_i)*p_j*(lt_i+lt_j)+p_i*(1-p_j)*lt_j$，两者作差得$i$ 在j前更优当且仅当$(1-p_j)*p_i*lt_i < (1-p_i)*p_j*lt_j$。 可以证明这个关系满足传递性，于是就可以排序了。

排完序之后就能开始dp了，排序的好处就是能保证这样做出来的期望罚时也是最少的。

令$dp_{i,j}$为二元组做了前i个题目花了j的时间得到的最大期望分数同时最小期望罚时。那么只要考虑$i+1$这个题目不做，做简单或全做三种情况。

转移很简单，这里只考虑全做时候的期望罚时，如果没有过就是原来的期望罚时加上简单部分的时间，如果过了，期望罚时就是时间j加上简单部分和困难部分的时间，然后分别加入期望罚时即可。

时间复杂度$O(nt)$，难度1/5。

Comment

由于期望分数要进行比较，就有一定的精度误差，题目中描述p_i不超过6位小数，可以把分数乘上10^6，然后用整数进行比较。

这个题也是挺有趣的，使用了现实中的比赛规则，并且也是一个挺有难度的题。

1.37 280D k-Maximum Subsequence Sum

Description

给定整数序列$a_1, a_2, \cdots, a_n$，执行m个操作，每个操作为以下两种之一：

- 把a_i赋值成val。
- 求$a_l, a_{l+1}, \cdots, a_r$中至多$k$段最大不相交子段和。

规定$n, m \leq 10^5, k \leq 20$。

Analysis

首先考虑k段不相交子段和，使用最大费用流解决，即源向每个点i连流量为1，费用为0 的边，然后i向$i+1$连流量为1，费用为a_i的边。每个点向汇连流量为1，费用为0的边，然后增广k次即可。

然后使用数据结构加速这个过程，经过一次增广之后，原来的正向边变成反向边，1到$n+1$的链上一定是正向边反向边交错的。一条可行路径就是正向边或反向边中的连续一段。每次增广的路径也是其中和最大的连续一段，增广之后原先一段又正向边或反向边连续一段分成三段，于是只要使用堆维护这些区间，线段树查询最大子段和即可。

这个过程是可以简化的，可以每次找出最大子段和，然后取反这样做k次，这样做一定能得到解，并且不会差于上述的解也就是最优解，因为全局的最大子段和不会差于若干个区间的最大子段和。所以这样也能得到最优解。

于是只要使用线段树维护最大子段和就行了，当然要支持取反操作。

时间复杂度$O(mk \log n)$，难度1/5。

Comment

通过和费用流做法的比较，能得出这样的结论，每次取的区间或者和之前的区间不交，或者是某个区间的子区间，直接说明这个是不容易的。

k段不相交子段和也是个很经典的问题，很早就有研究。这篇paper发表于2007年，讲了这个问题的线性做法，并且出现在Rockethon 2014中。

国内似乎早年出现在POJ challenge B Birthday Gift中，作者ftiasch给出的做法从小到大合并，用堆维护，与这个做法其实是一个逆过程。

1.38 283E Cow Tennis Tournament

Description

有一个n个数的序列$s_1, s_2, \cdots, s_n$，每个数两两不同。一个$n*n$的矩阵M定义为当$i \neq j$时，如果$s_i \leq s_j$，那么$M_{i,j} = 1$，否则$M_{i,j} = 0$。

有k个操作，每次选择两个参数$a, b(a < b)$，对所有i, j，如果$s_i, s_j \in [a, b]$，那么将$M_{i,j}$取反，即将0变成1，1变成0。

问最后有多少个三元组(p, q, r)，p, q, r两两不同，且$M_{p,q} = M_{q,r} = M_{r,p} = 1$。

规定$n, k \leq 10^5$。

Analysis

首先这就是竞赛图的三元环计数，通过补集转化就变成了求不形成三元环的三元组个数，通过一些简单的讨论，这三个点的导出子图与$p \to q, q \to r, p \to r$ 这样的图同构。也就是$\sum_{r=1}^{n} \binom{ind_r}{2}$，$ind_r$表示点$r$的入度。

而r的入度就是第r列1的个数，这个显然通过扫描线加线段树维护即可。

时间复杂度$k \log n$，难度0.5/5。

Comment

关于竞赛图三元环计数的技巧，可能十几年前就出现在算法竞赛中了。而一般任意图的三元环计数没有太好的方法，可以证明一个k条边的图三元环的个数是$O(k^{3/2})$的，并且在这个代价里枚举出来，更精确的上界是$\sqrt{2}k^{3/2}/3$，我在刷数竞时做到过这个题。

然而这个题又是一个无聊的拼题，实在令人捉急。

1.39 285E Positions in Permutations

Description

问有多少个$1 \sim n$的排列p，满足恰好k个位置有$|p_i - i| = 1$。

规定$n \leq 1000$，答案对$10^9 + 7$取模。

Analysis

Solution 1

用$dp_{i,j,k}$表示前i个数选了j个位置满足$|p_i - i| = 1$，k表示$i, i+1$是否出现p 中。

令F_j表示至少j个位置满足$|p_i - i| = 1$，$F_j = \sum dp_{n,j,k} * (n-j)!$，因为有$n-j$个数是未确定的。

接下来就是一个简单的容斥，算出恰有j个位置满足$|p_i - i| = 1$。

时间复杂度$O(n^2)$，难度1/5。

Solution 2

首先可以看成二分图匹配，若$|i - j| = 1, A_{i,j} = x$，否则$A_{i,j} = 1$，答案显然是$\mathrm{perm}(A)$中x^k前面的系数。其中$\mathrm{perm}(A)$表示A 的积和式。

积和式可以用如下方法计算，对每个集合$(e_1,e_2,\cdots,e_t)$其中$x_1,x_{e_2}\cdots$，x_{e_t}两两不同并且$y_{e_1},y_{e_2},\cdots,y_{e_t}$ 两两不同，在答案中加上$(n-t)!\prod_{i=1}^{t}(w_{e_i}-1)$。

可以用下面的式子证明：

$$\mathrm{perm}(A+B)=\sum_{s,t}\mathrm{perm}(a_{ij})_{i\in s,j\in t}\mathrm{perm}(b_{ij})_{i\in\bar{s},j\in\bar{t}}$$

其中s和t是$\{1,2,\cdots,n\}$中大小相同的子集，$\bar{s}$, $\bar{t}$是他们的补集。

构造一个无向图G有$2n$个结点$v_1,v_2,\ldots,v_{2n}$，v_i,v_{n+j}之间的边权为$A_{i,j}-1$。我们只要知道所有选择t 条边的匹配的权值的和即可。一个匹配的权值为所有在匹配中边的边权的乘积。

所以我们只要知道每个连通分量中选择x条边的权值和即可。

回到原题。对于所有非1的边，构成了两条长度为n的链。每条链中选x条边，等价于$\{1,2,\cdots,n\}$中选择大小为x的子集，使得数两两不相邻，所以方案数为$\binom{n-x}{x}$，所以一条链的生成函数为$f(x)=\sum_{i\geq 0}\binom{n-i}{i}(x-1)^i$。

所以整个图的生成函数为$f(x)^2$，这个两个部分显然都是一个卷积，所以可以使用FFT加速。

时间复杂度$O(n\log n)$，难度2.5/5。

Comment

我在当时一时没有想到算法1，于是想到了算法2。可以算法2当成算法1一个更数学性的描述。

算法2受到两个题目启发而得，TCO 2014 3B TreeDistance，求至少k条指定边的生成树个数，wata在这个题中把指定边的权值当成x，然后暴力代入1到n，然后插值求出原多项式，通过数学的手段避免了琐碎的动态规划。Codeforces 268E，这是我出的关于积和式计算的一个题，在和hos.lyric的交流中知道了关于积和式计算的一些技巧。

1.40 286D Tourists

Description

在一个直角坐标系中，有m条线段会在t_i时刻出现在$(0,l_i)$和$(0,r_i)$，并且不消失。

有n个询问，2个点在时刻q_i从$(-1,0)$和$(1,0)$出发，以1的速度向y轴正方向移动，有多少时间这两个点之间不可见，也就是两点连线至少与一条线段相交。

规定$n,m\leq 10^5$。

Analysis

首先把时间当成x轴，原坐标系的y轴当成y轴，建立直角坐标系，那么每条线段就代表$x\geq t_i,l_i\leq y\leq r_i$这样的一个范围。而询问就代表$y=x-q_i$这条直线与上面区域的交的长度。

这样直接处理是不方便的，因为每条线段代表的区域之间是有交的，所以我们只要搞成一个一个不交的区域就能分别计算贡献了。

所以只要求出每个位置最早什么时刻被覆盖，把相同的合并成一段，显然这样处理之后段数不会超过$2m$，具体的实现就是按时间从后往前排，然后每条线段将这一段之间染色，用平衡树或线段树维护一下即可。

对于一个区域$x \geq t_i, l_i \leq y \leq r_i$，那么对于时间$q < t_i - r_i$，那么贡献为0，而对时间$q \in [t_i - r_i, t_i - l_i]$，贡献为$q - t_i + r_i$，而对$q > t_i - l_i$，贡献为$r_i - l_i$，所以只要按时间点排序维护一个一次函数即可。

时间复杂度$O(n \log n)$，难度1.5/5。

Comment

这是一个还不错的题目。对于求斜线与这些区域的交，第一反应可能就是纵切变换，将(x, y)变成$(x - y, y)$，那么每条斜线就变成了直线，但这样那些区域处理的难度大大增加。刚开始做这个题的时候，我一直在想怎么维护这些区域。后来才发现要在原坐标系中处理完，再进行纵切变换算贡献，还是挺机智的一个数据结构题。

1.41 286E Ladies' Shop

Description

给定集合$A = \{a_1, a_2, \cdots, a_n\}$。

定义函数$f(S)$表示S内所有数的和，$g(S) = \{x | x \in S\}$

问能否找到一个集合$P = \{p_1, p_2, \cdots, p_k\}$，满足：

- 对于A中每一个数a_i，存在可重集S，其中$g(S) \subseteq P$且$f(S) = a_i$。
- 对于任意可重集S，其中$g(S) \subseteq P$且$f(S) \leq m$，都有$f(S) \in A$。

并且要求集合P尽量小。

规定$n, m \leq 10^6$，且$1 \leq a_i \leq m$。

Analysis

首先考虑是否存在可行的P，这个等价于A是否可行。如果A可行，那么令$P = A$即可，如果A不可行，那么存在A中一个可重集和不超过m，但总和不在A中，由条件1得A中每个元素都是P的一个可重集的和，所以这个总和也能表示成P中某些元素的和，这就矛盾了，所以P也不可行。

那么问题就转化成A是否可行了，条件1显然是满足的，关于条件2，如果A不可行，即Q 为A中最小的可重集，且满足$f(Q) \leq m, f(Q) \notin A$，那么如果$|Q| \geq 3$，令$Q = \{q_1, q_2, \cdots, q_{|Q|}\}$，又最小性得$f(Q \backslash q_1) \in A$且不超过$m$，那么令$f(Q \backslash q_1) = a_k$，于是就有$\{a_k, q_1\}$满足条件，与$Q$的最小性矛盾。如果$A$ 不可行则存在$a_i + a_j \leq m$且$a_i + a_j \notin A$。反之如果如果A是可行的，那么显然对所有$a_i + a_j \leq m$，都有$a_i + a_j \in A$。这也是A可行的充要条件。

这个判断显然可以构造多项式$f(x)=\sum x^{a_i}$，然后求$f(x)^2$就能得到a_i+a_j的所有项，这个可以用FFT做。

然后要求P集合最小。由条件2得$P\subseteq A$。如果一个数a_k能表示成a_i+a_j，那么可以将a_k从P中删去，因为p_k一定能表示成若干个P 中数字的和。否则这个数字一定要在P中，不然a_k不能被表示成其他数字的和，与条件1矛盾。

时间复杂度$O(m\log m)$，难度1/5。

Comment

这个题的条件很奇怪，我们要做的是充分挖掘条件的性质。如果从结论说明为什么成立，这是很显然的，而上面的题解则是从条件出发，以一个正常人的思维，一步一步推出结论。如果是不正常人直接看出结论也是有可能的。

这场Round还是很赞的，虽然也不是太难，但每个题目都很有趣的，无论从题目的设定还是做法上。一般来说，题目的难度和有趣程度是负相关的，如果能出一个又有难度又有趣的题，那么作者的功力应该是很深厚的。作者叫tunyash，在CF 上出了四场，每场题目都很不错，但是这次作业中只被选入了一场。

1.42 288E Polo the Penguin and Lucky Numbers

Description

定义幸运数字为十进制中只包含4和7的正整数。

有两个幸运数组$l,r(l<r)$，假设$[l,r]$中有n个不同的幸运数字，从小到大为$a_1,a_2,\cdots,a_n$，求$\sum_{i=1}^{n-1}a_i*a_{i+1}$。

规定$l<r\le 10^{100000}$，答案对10^9+7取模。

Analysis

数位DP，要解决的问题是从$\sum_{i=1}^{n-1}a_i*a_{i+1}$求$\sum_{i=1}^{n-1}(a_i+x)*(a_{i+1}+x)$，那么展开之后就维护一下所有数的和就可以了。

所以问题解决了。

时间复杂度$O(\log r)$，难度0/5。

Comment

这是witua前辈最后一场CF，然后一位叫sereja的前辈扛过了他的大旗，接着在CF上出着一场又一场有趣的比赛。

1.43 293B Distinct Paths

Description

有一个$n*m$的木板，一些块已经被涂上给出的k种颜色中的一种。你需要把每个没涂色的块涂色使得从左上角到右下角的每条路径都不会经过两个颜色一样的块。路径只能向右或向下走。

规定$n,m\le 1000,k\le 10$，答案对10^9+7取模。

Analysis

首先发现路径的长度是$n+m-1$，而颜色只有k中，所以如果$n+m-1>k$，那么答案一定为0。

那么木板的大小不会很大，搜索即可。搜索过程中，即时维护每个块能涂的颜色。并且只要搜出本质不同的涂色方案，然后乘上组合数即可。

难度1/5。

Comment

这个题除了搜索之外看上去没有靠谱的状压DP做法，并且我也不会一些复杂度靠谱的搜索。如果真的是这样，那出这样的搜索题也是挺不靠谱的。

1.44 293D Ksusha and Square

Description

有一个凸n边形，顶点为$(x_i, y_i)(i = 1, 2, \cdots, n)$，在其内部(包括边界)随机选择两个不同的格点，求以这两个格点连线为对角线的正方形面积期望。

规定$n \le 10^5, |x_i|, |y_i| \le 10^6$。

Analysis

我们要知道格点总数和面积总和。对于格点(x_1, y_1)和(x_2, y_2)，那么面积为$\frac{1}{2}((x_1-x_2)^2+(y_1-y_2)^2)$。那么显然$x$和$y$坐标能分开来算。

假设所有格点的x坐标为$x_1, x_2, \cdots, x_m$，那么有

$$\sum_{i<j}(x_i - x_j)^2 = m\sum_{i=1}^{m} x_i^2 - (\sum_{i=1}^{m} x_i)^2$$

所以只要求$\sum_{i=1}^{m} x_i^2$和$\sum_{i=1}^{m} x_i$即可。

Solution 1

注意到坐标范围很小，所以只要暴力，并用指针维护当前上下边界在哪条边即可。

记W为坐标范围，时间复杂度为$O(W+n)$，难度0.5/5。

Solution 2

可以看成$O(n)$段，每段都是两条直线之间的点，那么就一定能表示成$\sum i[\frac{ai+b}{c}]$和$\sum i^2[\frac{ai+b}{c}]$ 的形式，于是可以使用欧几里得算法。

记W为坐标范围，时间复杂度为$O(n\log W)$，难度1/5。

1.45 293E Close Vertices

Description

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。

统计有多少个点对(u, v)，u, v之间路径长度不超过L，权重不超过W。

规定$n \le 100000$。

Analysis

点分治，然后可以套用任何$O(n \log n)$的二维数点做法。

时间复杂度$O(n \log^2 n)$，难度0.5/5。

Comment

An old-fashioned data structure problem.

1.46 294D Shaass and Painter Robot

Description

有一个$n * m$的网格，一开始都是白色的。有一个机器人，一开始位于边界的某个格子(x, y) 上，面朝一个对角线方向，顺着这个方向走下去，每当遇到边界时会遵循光的反射定律改变方向，求将这个$n * m$的网格变成黑白相间需要走多少步，或者指出不可能。

规定$n, m \le 10^5$。

Analysis

首先我们要注意到，整个图被黑白染色当且仅当所有边界格子被黑白染色时。

必要性显然，下证充分性。如果边界上的格子都被黑白染色，且中间有一个黑格子没被染成黑色。这个格子对应着两条对角线，因为这个格子没被染成黑色，路径中不包括对角线，然而这两条对角线对应的四个边界上的格子都被染黑，也就是一条路径上有四个度为1的点，矛盾。

所以只要模拟机器人的路径，然后维护边界上的格子染色情况即可。如果进入一个循环且边界上的格子没有被染完那么就是无解的。

时间复杂度$O(n + m)$，难度0.5/5。

Comment

这个题作为div 2的D题还是有点难的，因为我并不觉得这个结论很显然，当然也和我一开始没看到机器人在边界上有关。如果机器人不在边界上，那么还要维护一下机器人出发位置所在两条对角线格子的染色情况。

1.47 295D Greg and Caves

Description

有一个$n * m$的网格，每个点为黑色或白色。

问有多少种染色方式满足下面条件:

- 存在区间$[l,r]$，使得l到r之间每一行恰有两个黑点，而其他列只有白点。

- 存在$t \in [l,r]$，使得在$[l,t]$之间左边黑点所在列的编号不增，右边黑点不减，$[t,r]$之间左边黑点所在列的编号不减，右边黑点不增。

规定$n,m \le 2000$，答案对10^9+7取模。

Analysis

首先发现，区间的具体位置没什么用，只要知道长度就行了。

$f_{i,j}$表示做了i行，当前两个黑点中间有j个格子的方案数，那么$f_{i,j} = \sum_{k=0}^{j} f_{i-1,k} * (j-k+1)$，这个使用前缀和加速即可。

然后枚举中间行t和间距j，为了不重复，令t为满足条件最大的行，即$t+1$行两个黑格间距小于j，于是只要将两部分乘一下即可，同时乘上$m-j+1$，即可以摆的位置，加入答案。

时间复杂度$O(n^2)$，难度0.5/5。

1.48 297E Mystic Carvings

Description

一个环上有$2n$个点，n条线段将这些点一一配对。

求线段的三元组个数，使得这些线段两两相交或两两不相交。

规定$n \le 10^5$。

Analysis

把相交的线段之间连黑边，不相交连白边，那么显然就是同色三角形计数。可以使用补集转化，那么要统计的只是与每条线段相交的线段数目。

与一条线段$[l,r]$相交的线段$[a,b]$，那么有$a \in [1,l), b \in [l,r]$或$a \in [l,r]$，$b \in (r,2n]$，那么使用任意二维数点的方法就能解决。

时间复杂度$O(n \log n)$，难度0.5/5。

1.49 301C Yaroslav and Algorithm

Description

设计一个程序，程序有若干条命令组成，第i条命令为"s_i>>w_i"或"s_i<>w_i"，其中s_i, w_i 是有数字或"?"组成的长度不超过7的字符串。

这个程序每次读入一个字符串a，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，那么整个程序终止。

假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"，那么将新串作为程序的读入继续执行。否则，程序终止。

输出即为终止时的字符串。

请设计程序能对一个读入的数字串实现加1功能。

Analysis

第一部分就是在串末尾加入标记，这个只要在开头加入标记“??”，然后将标记不停往后移即可，具体做法是“??x>>x??”，其中x是任一数字。

接下来就是考虑加一和进位，为了区分两个过程，将“??”改为“?”，如果“?”前面的数不为九，那么只要将前面的数加一即可。否则将“9?”改为“?0”，接着执行。注意如果“?”来到了串头将它变为“1”。

当然为了防止前面的过程交叉，需要适当调整一下命令的顺序。

难度0/5。

Comment

曾经有一段时间我一直以为“?”是通配符，直到我膝盖中了一箭。

推荐可以玩一下manufactoria和hacker.org里面的hvm。

1.50 301E Yaroslav and Arrangements

Description

数列$a_1, a_2, \cdots, a_r$是良好的，当它满足：

- 对所有$1 \le i \le r$，$|a_i - a_{i+1}| = 1$，其中记$a_{r+1} = a_1$。
- $a_1 = \min_{i=1}^{r} a_i$。

数列$b_1, b_2, \cdots, b_r$是优秀的，当他满足：

- 数列中的元素不降。
- 满足$1 \le r \le n, 1 \le b_i \le m$。
- 重排数列得到至少一个至多k个不同的良好数列。

给定n, m, k，求有多少个优秀数列。

规定$n, m, k \le 100$，答案对$10^9 + 7$取模。

Analysis

考虑构造一个良好的数列，最简单的良好数列就是由12循环产生的数列。然后加入3那么显然是在某对12之间插入23变成1232，假设有a 对12插入了b个23，那么方案就是$\binom{a+b-1}{b}$。加入34同理，并且两个过程独立的，所以只要相乘即可。

令$dp_{i,j,a,b}$表示当前最大数是i，现在数列中有j，方案数是a，前一个数出现了b次。容易证明不超过k的组合数是$O(k)$的，所以当前三维固定时，最后一维转移均摊是$O(1)$的。

时间复杂度$O(k^4)$，难度0.5/5。

Comment

容易看出，这是无机结合的计数题，也是sereja的计数题常见的风格。

1.51 303E Random Ranking

Description

有n个变量，第i个变量随机独立分布在$[l_i, r_i]$之内，问对所有i, j，第i个变量是第j大的概率。

规定$n \leq 80$。

Analysis

首先可以离散化出$O(n)$个区间，可以对每个区间分别计算变量i在这个区间内第j大的概率，然后相加。

在区间$[l, r]$内，可以算出变量i有a_i的概率在$[0, l)$之间，即区间左边，b_i的概率在$[l, r]$之间，c_i的概率在(r, ∞)之间，即区间右边。那么记$P(i) = a_i x + c_i y + b_i$，那么将所有$P(i)$乘起来得多项式$F(x, y)$，$x^p y^q$前面的系数即为有$p$个数在左边$q$个数在右边的概率，而剩下的数在中间均匀分布。所以得到名次是$p+1$到$n-q$之间的概率是均等的。

所以对于变量i，$F(x, y)/P(i)$就是除了i之外其他数的分布，于是可以算出它本身的概率。并且这样一次多项式除法是$O(n^2)$的，每个区间做$O(n)$次。

时间复杂度$O(n^4)$，难度1/5。

Comment

这样做虽然很好，但是要用到除法，所以会挂精度。解决的办法就是改成$O(n^5)$，除法用暴力乘起来代替，这样需要一点常数优化。或者是改成分治，对于区间$[l, r]$计算出这个区间内所有数除了本身的乘积，那么先分治下去，然后将右边整体的乘积乘到左边每一个上，左边同理。这样复杂度是$O(n^4 \log n)$的。

不过我觉得这个题作为Chinese Round的E题还是有点简单了。

1.52 305D Olya and Graph

Description

有一个n个点m条边的有向图，问有多少种方案添加任意数量的边，使得改图满足下列条件:

- 从点i出发，可以到达点$i+1, i+2, \cdots, n$。
- 任意从u到v的有向边都有$u < v$。
- 两点之间最多有一条边。
- 对于一对点$i, j (i < j)$，若$j - i \leq k$，那么从i到j的最短路等于$j - i$。
- 对于一对点$i, j (i < j)$，若$j - i > k$，那么从i到j的最短路等于$j - i$或$j - i - k$。

规定$1 \le n,k \le 10^6, m \le 10^5$，答案对$10^9+7$取模。

Analysis

由1和2得i和$i+1$之间一定存在一条边。

如果存在一条边$i, i+u$，其中$1 < u \le k$，那么与条件4矛盾。如果$k+1 < u$，那么与条件5 矛盾。

所以除了i到$i+1$的边之外只能存在i到$i+k+1$的边，记这样的i形成的集合为A。

如果A中存在两个数i, j，$j - i > k$，那么i到$j+k+1$的最短路不超过$j-i-2k$矛盾。

所以A中集合差不超过k，那么只要枚举最小的数，然后A中有哪些数字可以选即可。

时间复杂度$O(n)$，难度0/5。

Comment

容易发现，稍加优化就能做到$O(m)$，与读入同阶，但是没啥必要。

1.53 305E Playing with String

Description

有一个字符串集合，两个人轮流操作，不能操作的人输。

每次操作选择一个字符串t，找到一个位置$1 < i < |t|$，满足$t_{i-1} = t_{i+1}$，然后将字符串分成三段$t_{1\sim i-1}, t_i, t_{i+1\sim |t|}$。

游戏开始时只有一个字符串s，问先手是否有必胜策略，并输出最小可行策略。

规定$|s| \le 5000$。

Analysis

一个直接的想法就是$dp_{l,r}$表示这一段的SG值，然后枚举分割点计算。但这样时间复杂度是$O(|s|^3)$，需要进一步的分析。

将$1 < i < |s|$且$t_{i-1} = ti+1$的位置涂黑。那么每次操作就相当于找一个黑点，然后将这个格子和周围两个格子全都变白。

这样转化之后就能发现两段不相连的黑色连续段之间是独立的。于是只要令dp_i表示连续i个黑色格子的SG值，这个时候只要枚举中间点就行了。

时间复杂度$O(|s|^2)$，难度0.5/5。

Comment

这个题虽然不难，但还是挺精致的。

1.54 306C White, Black and White Again

Description

在n天内发生了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。并且n天中先有若干

天发生好事，再有若干天发生坏事，再有若干天发生好事。求方案数，其中每天发生的事有顺序。

规定$n, w, b \le 4000$，答案对10^9+9取模。

Analysis

首先不考虑事件的顺序，只要在答案最后乘上$w!b!$即可。

假设i天发生了好事，那么$n-i$天发生了坏事，发生坏事的方案数为$\binom{b-1}{n-i-1}$，好事的方案数为$\binom{w-1}{i-1} * (i-1) = \binom{w-2}{i-2} * (w-1)$，即将坏事插入好事的方案数。所以答案为$w!b!(w-1)\sum_{i\ge1}\binom{b-1}{n-i-1}\binom{w-2}{i-2} = w!b!(w-1)\binom{w+b-3}{n-3}$。

时间复杂度$O(n)$，难度0/5。

1.55 306D Polygon

Description

构造一个n条边的凸多边形，使得每个角相等，每条边不等。

规定$n \le 100$。

Analysis

构造一个正n边形，然后将每条边沿法向量随机平移一段距离。

时间复杂度$O(n)$，难度0/5。

Comment

由于10^{-3}的误差算相等，所以需要精心地调参数。

1.56 309B Context Advertising

Description

有n个单词，选最长的连续一段，使得能放入一个$r*c$的表格中，要求表格中一行内单词之间用空格隔开，一个单词必须在同一行内，且不能改变单词的顺序。

固定$n, r*c \le 10^6$，总长度不超过$5*10^6$。

Analysis

首先使用two pointer可以求出以这个单词为这一行的开始，下一行的开始为哪个单词。

然后使用倍增即可。

时间复杂度$O(n\log c)$，难度0.5/5。

1.57 309D Tennis Rackets

Description

边长为n的正三角形，被$n-1$个点均匀分开。每条边上在离顶点距离超过m的点中选一个，求多少种方案能构成钝角三角形。

规定$n \le 32000$。

Analysis

枚举钝角所在的点，然后再枚举另外一个点，做一下垂直就能得到另外一条边上能选的点的范围。

时间复杂度$O(n^2)$，难度0.5/5。

Comment

虽然算法很简单，但是需要一些常数优化，首先由于对称性，所以枚举钝角对应的点只要枚举到$n/2$即可，然后当枚举另一个点的时候，另外一条边上可取的范围首先是全部，然后是中间的一部分，然后就没有了。两端的两部分可以快速计算，然后中间的范围随着那个点的移动而越来越小的，这个时候可以使用two pointer维护而不是直接做垂线，这样就避免了除法。而在two pointer 维护过程中判断是否为钝角的式子可以稍微进行一下简化以减少运算的次数。所以这就是一个痛苦而无聊的常数优化题。

1.58 311C Fetch the Treasure

Description

有h个房间，其中n个里面有宝藏，第i个宝藏房编号为a_i，里面的价值为c_i。

一开始从1号房间出发，每次只能往前走k步。

然后有一下m个操作，每个操作是下列三种类型之一:

- 增加技能x。即允许Freda每次前进x步。
- 让第x间宝藏房的价值减少y美元。
- 查询能到达的房间中价值最大的宝藏并拿走。如果能到达宝藏房且有多个宝藏房里的价值达到最大，那么取走宝藏房编号最小的宝藏密室。

规定$h \le 10^{18}, n, m \le 10^5, k \le 10^4$，且1操作个数不超过20个。

Analysis

首先第二和第三个操作是平凡的，只要维护能到的宝藏房的集合即可。

考虑第一个操作，首先将所有集合划分成模k的同余系，那么在一个等价类中，如果x可达，那么$x+k$可达，所以只要知道可达的最近的格子即可。

令dp_i表示能到达的模k等于i最近的格子，那么对于技能x，将$dp_i + x$更新$dp_{(i+x) \bmod k}$，这就是求最短路的松弛操作，所以只要跑一边最短路就行了。

由于这个图的特殊性，是若干个环，并且边权相同，所以只要在环上找权值最小的点，然后从这个点开始更新一遍整个环即可。由这个似乎就能得到实现好的队列优化bellman ford算法也是线性的。

操作1时间复杂度$O(k)$，操作$2,3$时间复杂度$O(\log n)$，难度0.5/5。

Comment

我已经在超过两个地方见过这个题了，一是周奕超2011年集训队互测题墨墨的等式，然后是codechef MARCH13 LECOINS，还有是POJ某题，具体题号找不到了。

经过我进一步的考证我发现这好像是2002的WC题《牛场围栏》，这实在令我吓了一跳。

不过我觉得这种题出过一次之后就没有什么意思了。

1.59 311E Biologist

Description

有n个变量，一开始有初始值0或1，可以用v_i的代价改变第i个变量的值。

并有m个约束，每个约束是一个k_i大小的集合和c_i，当集合中所有的变量都为c_i时，就有w_i的收益，否则有g_i的亏损。

求能得到的最大收益。

规定$n \le 10^4, m \le 10^3, k_i \le 10$

Analysis

考虑最小割，和源连表示变量取0，和汇连表示变量取1，然后开始凑各个限制。

首先如果第i个变量初始值为0，那么由源向这个点连费用为v_i的边，表示改变这个的代价，为1同理。

对于约束i，如果$c_i = 0$，那么从源向约束i连$w_i + g_i$的代价，表示这个限制不满足的代价。然后由约束i向其中所有变量连无穷大的边，表示如果限制和源相连，那么所有变量都要和源相连，即取0，$c_i = 1$同理。

那么所有约束的收益相加并减去最小割即得最大收益。

时间复杂度$O(\text{maxflow}(n+m, n+m+\sum_{i=1}^{m} k_i))$，难度0.5/5。

Comment

这个建模显然是很简单的，只要猜出和源汇连的意义，然后凑就行了，更难的类似的建模可以看srm 558的hard题。

这场是Chinese Round，题目质量比较差，都是原题或者典型的OI题，没有啥有意思的题。

1.60 314E Sereja and Squares

Description

有n个点，每个点被标上了除了“x”以外的小写或大写英文字母。要求这些点被分成若干对，每对中编号小的点标小写字母，编号大的标大写字母，而且每对点对应的区间不交叉。

现在这些点部分被标上了小写字母，可以对未标的点标字母，并满足上述条件。问有多少种标号方案。

规定$n \le 10^5$，答案对2^{32}取模。

Analysis

首先要认识到这就是一个括号序列，小写字母等价于左括号，大写字母等价于右括号。假设有m个括号被标了字母，那么剩下$n/2-m$个括号都是可以随便标的，于是只要乘上一个25的幂次即可。

关于括号序列就是$dp_{i,j}$前i个字母，j层左括号的方案数，因为i和j的奇偶性相同，且$j \le i, j \le n-i$，所以实际的状态就是$n^2/8$，然后加一点常数优化就能过了。

时间复杂度$O(n^2)$，难度0/5。

Comment

sereja将一个简单的$O(n^2)$dp，出到10^5，并放到E题，使通过人大大减少。这就是一个出题人的水平和高度，有化腐朽为神奇的力量。我想换个人出这个题难度可能就大打折扣了。

经过多次教训，我终于知道了关于括号匹配的题，无论求方案还是求最大值，无论范围多大，一些看上去复杂度很差的做法直接上。

1.61 316E Summer Homework

Description

有一个数列$a_1, a_2, \cdots, a_n$，要进行m次操作，每个操作为如下三种之一：

- 将a_x赋值为v。
- 求$\sum_{x=l}^{r} f_{x-l} * a_x$，其中$f_0 = f_1 = 1$，对于$x \ge 2$，$f_x = f_{x-1} + f_{x-2}$。
- 将下标在$[l, r]$之内的数加上d。

规定$n, m \le 10^5$，答案对10^9取模。

Analysis

通过数学归纳法可得$f_{x-l} = (-1)^l (f_l * f_x - f_{l-1} * f_{x+1})$。于是只要维护$\sum f_x * a_x$和$\sum f_{x+1} * a_x$即可。

那么剩下来的工作就是显然的线段树了。

时间复杂度$O(m \log n)$，难度0.5/5。

Comment

并不是所有的递推数列都能方便的找到类似f_{x-l}这样的关系式的，于是可以考虑利用特征根求出通项或者直接暴力矩阵。

1.62 316G Good Substrings

Description

给一个字符串s和n个字符串t_i，问s中有多少个不同的子串满足在t_i中出现了$[l_i, r_i]$次。

规定$|s|, |t_i| \le 5 * 10^4, n \le 10$。

Analysis

首先将所有串连起来，建出后缀树。在树上统计每个结点在每个子串中出现几次就是一个简单树形dp。

时间复杂度$O((|s| + \sum |t_i|)|\Sigma|)$，$|\Sigma|$为字符集大小，难度0.5/5。

Comment

此题清澄上内存限制比较紧，所以在建树的时候儿子直接用map存，这样空间复杂度就与字符集大小无关了。

1.63 317C Balance

Description

有一个n个点m条边的图，每个点有个权值w_i，满足$0 \le w_i \le v$。一次操作为选择一条边(a, b)，选择一个数c，然后是w_a加上c，w_b减去c，但要保证$0 \le w_a, w_b \le v$。

给定初始权值a_i和目标权值b_i，求一个不超过操作$2n^2$之内的方案，达到目标。

规定$n \le 300$。

Analysis

首先猜想每个连通分量里面的数字和相同，那么一定是有解的。接着给出一种构造方法。

如果一个连通分量有解，那么它的一个生成树也是有解的，所以可以忽略其他边。接下来要做的就是把点一个一个地调整到目标，然后从图中删去，为了保证图的连通性，每次从叶节点开始做。

现在做结点u，那么不妨设$a_u < b_u$，否则令$a'_u = v - a_u, b'_u = v - b_u$，与原问题等价。也就是差$d = b_u - a_u$的权值。从结点$u$开始深度优先遍历这棵树，将遍历到结点的权值加入总和中，直到超过d为止，那么将这些权值沿着树边流向u中，因为过程中结点的权值不会超过总的权值d，所以不会超过限制。于是问题就解决了。并且操作次数不超过$n * (n - 1)/2$。

时间复杂度$O(n^2)$，难度0.5/5。

1.64 317E Princess and Her Shadow

Description

有一个二维平面，m个格子内有障碍。起始A和B在两个不同的格子上。每一次，A向一个没障碍的格子移动，然后B也向同一个方向移动，但是如果对应的方块有障碍，B不能移动。

构造移动方案，使得不超过10^6次移动后A和B能在同一个位置上。

规定$m \le 400$，且障碍的横纵坐标绝对值不超过100。

Analysis

首先一个显然的事实就是如果A和B在两个不同的连通块中或者一个障碍都没有就显然是无解的。

如果A和B是不封闭的，也就是A和B能到达的格子无限多。首先定义能包含障碍的最小的与坐标轴平行矩形所在的区域为里面，那么可以让A沿着路径先走到外面，并且走得足够远。如果B在里面，那么让B也沿着路径走到外面。因为A足够远，那么肯定不会走回里面。所以此时AB都在外面。接下来让A去接近B，如果A在B的下方，那么让B走到最下面的障碍下面一格，然后A往上走即可。这样B 固定不动，直到AB的纵坐标相同。其他方向同理，所以只要这样做AB就能到同一个位置上。

如果A和B在封闭区域中，那么首先找到A和B的最短路径，然后A沿着最短路径往B走，不断重复这个过程。正确性只要对AB之间最短路径长度进行归纳即可，如果B在过程中碰到障碍，那么AB之间的最短路径减1。如果B过程不碰到障碍，那么每次都沿着同一方向移动，所以走有限次就会碰到边界，矛盾。

难度2/5。

Comment

挺有趣的构造，从代码和思路上来说都不简单。虽然是个原题，但是见过的人应该不多。

此题封闭部分出自2012 年加拿大数学竞赛第4题，不过未找到官网上的相关链接。

同时我也玩过类似的小游戏，操控若干个东西同步移动，然后达到一一对应的位置，不过这个游戏看上去是没有通用的构造方法的。

此题不封闭的部分这样做步数的上界是很好证的，但是封闭部分直接估计应该是会超过10^6步，所以对上界需要一些精确的估计，然而我并没有做。

1.65 319D Have You Ever Heard About the Word?

Description

一个字符串的重复块为能表示成ww的形式的子串，其中w为一个字符串。

有一个由小写英文字母组成的字符串S。每一步找到它的子串中最短的重复块，如果有多于一个，你必须选择最左边的那个。并将那个形如ww的重复块替换成w，重复以上步骤直到字符串中不存在重复块。

规定$|S| \le 5 * 10^4$。

Analysis

首先要证明一个结论，每次删的重复块长度不降，相同长度的重复块，一定是从左到右。也就是删去一个块之后不会出现更短的重复块也不会在左边出现长度相同的重复块。

要产生重复块，那么一定是要跨过整块。否则假设pw中产生重复块，那么pww中也有，那么当前删去的一定不是w，矛盾。重复块出现在wp中同理。所以重复块一定是$pwwq$删去w之后形成的pwq，其中$|p|, |q| \geq 1$。如果重复块与ww长度相同，那么w一定能被表示成qp，即$pwq = pqpq$，此时$pwwq = pqpqpq$，当前应删去pq，与w矛盾。如果重复块长度小于ww，那么$w = ab$，其中$pa = bq$且$|a|, |b| \geq 1$，否则w在重复块某一部分中，与长度小于ww矛盾。因为$|p| + |a| < |w| = |a| + |b|$，所以$|p| < |b|$，于是就有$b = pr, a = rq, pa = prq = bq$，那么有$pwwq = prqprrqprq$，此时$rr$是重复块且$|r| < |w|$，矛盾。证毕。

此时只要从小到大枚举重复块的长度，然后从左到右判断是否存在重复块删去即可。这样做时间复杂度是$O(|S|^2)$的，加上一些常数优化能在Codeforces上通过。

进一步优化这个做法，考虑长度为l的重复块，将这个串按长度为l的切成若干段。那么这个重复块一定会被完整的三段abc包含，且一定是$b = pq$，然后a的后缀为q，c的前缀为p，所以有重复块的充要条件就是ab的最长公共后缀与bc的最长公共前缀的和不小于l。找到之后将前面部分标记为删除，接着做下去即可。最长公共前缀可以只用hash+二分解决。除去删除部分剩下部分的时间复杂度是$O(|S| \log |S|/l)$的，所以总代价不超过$O(|S| \log |S|) * \sum_{i=1}^{|S|} 1/i = O(|S| \log^2 |S|)$

如果一个长度有重复块并且删去了，需要更新一下hash数组，但这样的不同长度显然不会超过$O(|S|^{0.5})$，所以这部分代价不会超过$O(|S|^{1.5})$。

时间复杂度$O(|S|^{1.5} + |S| \log^2 |S|)$，难度2/5。

Comment

我觉得这是一个很难的题目，尤其是刚开始的一个结论，并不显然，但却是做这个题的关键。但在我看到的许多题解中，把开始的这个结论当成一个显然的事实，并没有加上证明，而且现场也有很多人过了这个题目，这令我感觉挺奇怪的。最后字符串处理部分也不算太简单。

不过这个题CF上时限是6s，作者也没写题解，平方算法能过是不是作者的本意也无从得知了。

1.66 319E Ping-Pong

Description

维护一个区间集合，进行一下m个操作，每个操作为以下两种中的一种：

- 加入区间(x,y)，满足区间(x,y)的长度严格大于前面加入的任何区间。
- 询问能否从第x个区间移动到第y个区间，每次可以从集合中的区间(a,b)移动到另一个满足$c<a<d$或$c<b<d$ 的区间(c,d)。

规定$m \le 10^5$。

Analysis

如果将区间当成一个结点，先考虑一下这个图的连通性。首先如果两个区间相交，那么就是能互相到达，把这些在同一个强连通分量里面的线段合并，可以得到一个这个其中每个线段能到达的最左端和最右端，记作l,r。在不同强连通分量之间对应的区间或者不相交，或者包含。若两个强连通分量对应的区间相交，那么一定坐在两个小区间相交，矛盾。

如果一个强连通分量对应的区间被另一个包含，那么一定能从这一个到达另一个，否则不能。当然还有一个特殊情况，两个区间相等，即一个强连通分量由这个区间本身构成，另一个是由很多个小区间构成的，那么只能从小区间到达这个大区间。

接下来考虑如何维护这个结构，把区间合并起来求最左最右显然可以通过并查集解决。插入一个区间，如果它被某个大区间包含，因为区间长度递增，所以这个大区间中一定有一个小区间与这个大区间相交，所以可以直接并入这个大区间中。否则找到所有和它相交的区间，如果它和某个大区间相交，那么一定和其中某个小区间相交，所以可以直接合并，将原来的区间删除并加入新的大区间。至于如何维护这些区间，支持插入一个区间，删除一个区间，找到所有和某个区间相交的区间，可以使用树套树解决。

时间复杂度$O(m\log^2 m)$，难度1/5。

1.67 323B Tournament-graph

Description

构造一个有n个结点的竞赛图，对于任意两个结点最短距离不超过2。

规定$n \le 1000$。

Analysis

如果得出了x的解，那么可以构造出$x+2$的解，就是加上$x+1,x+2$这两个点，$1\sim x$的点向$x+1$连边，$x+1$向$x+2$连边，$x+2$向$1\sim x$连边。容易验证新的图也满足条件。

当$n=3$时显然是有解的，当$n=4$是无解，当$n=6$是有解，这两点可以通过爆搜得到，所以除了$1,2,4$无解，其他正整数都是有解的。

时间复杂度$O(n^2)$，难度0.5/5。

Comment

这题如果想到了递归构造就不难，但是由于奇数可以直接构造，所以会去尝试直接构造偶数，然后就做不出了。

这个题构造的验证是$O(n^3)$的，差于构造的复杂度。不过当然这是矩阵乘法，可以得到更低的复杂度。

1.68 323C Two permutations

Description

有两个包含n个元素的排列p和q，和m个询问，每次询问在p中位置在$[l_1, r_1]$,在q中位置在$[l_2, r_2]$中的数的数量。询问强制在线。

规定$n \le 10^6, m \le 2 * 10^5$。

Analysis

将每个数当成一个点，横纵坐标分别为p, q排列中的位置，询问等价于二维数点，可以使用可持久化线段树解决。

时间复杂度$O((n+m)\log n)$，难度0/5。

1.69 325C Monsters and Diamonds

Description

有n个怪兽，m个规则。第i个规则表示可以分裂成l_i个怪兽以及至少一个的钻石。

问从每个怪兽使用规则分裂，直到怪兽分裂完为止，最少最多得到的钻石是多少，或者说明无法全部分裂完，或者说明能分裂完但是能得到无限多的钻石。

规定$n, m, \sum l_i \le 10^5$，如果答案超过314000000且不为无穷大，用此数代替。

Analysis

首先考虑最少钻石，使用类似于最短路的做法。用优先队列维护当前可以被分裂完的怪兽，那么首先把所有存在规则分裂后没有怪兽加入优先队列。每次选取最小的确定答案，如果一个规则中所有的怪兽都确定了，可以把这条规则所分裂的怪兽也加入优先队列。同时这样就求出了无法分裂完的怪兽。

然后考虑最多钻石，首先把所有无法分裂完的怪兽和包括无法分裂完怪兽的规则删去。对于剩下的怪兽，向它能分裂出的怪兽连边，如果它在一个环内，那么它能通过不停的分裂得到自己，然后得到无穷多的钻石。于是剩下的怪兽构成了有向无环图，按着拓扑序倒序就能求出每个怪兽能分裂出最大的钻石数。

时间复杂度$O(n\log n)$，难度1/5。

Comment

此题好像还出现在江苏2014年省选中。

1.70 325E The Red Button

Description

有n个点编号为0到$n-1$，从第i个点能到达$2i \bmod n$和$(2i+1) \bmod n$两个点。

问能否从0号点出发，经过每个点恰好一次然后回到0号点。

规定$n \le 10^5$。

Analysis

首先考虑n为奇数，对于点y,$2x \equiv y \pmod n, 2x+1 \equiv y \pmod n$都有唯一解，也就是每个点恰有两个点指向这个点，0和$(n-1)/2$指向0，$n-1$和$(n-1)/2$指向$n-1$，于是就有$(n-1)/2$指向$n-1$和0，矛盾，所以奇数是无解的。

当n为偶数时，x和$x+n/2$[1]同时指向$2x$和$2x+1$这两个点，这就说明这两个指向的后继结点可以交换。一开始可以让这些点之间任意选择，每个点只有一条出边，一条入边，于是形成了若干条环。考虑x和$x+n/2$不在同一个环中，那么交换x与$x+n/2$ 的后继结点能将这两个环合并。于是可以将所有x和$x+n/2$并入同一个环中。

考虑0所在的环，对其中任意数x，$x+n/2$在环中，于是$2x$和$2x+1$也在环中，考虑每个数的二进制表示即得所有数都在环中。

时间复杂度$O(n)$，难度2/5。

Comment

可以避免使用并查集维护从而得到线性的复杂度。

这个题是个很精致的构造题，主要要注意到两个点之间可以交换的性质，然后就可以进行调整了。

然而这个题不加调整类似于贪心的做法，正确性不明，而且比赛的时候也有很多人用这种方法过掉了，不排除两种做法等价的可能性。

1.71 329D The Evil Temple and the Moving Rocks

Description

在一个$n*n$的正方形房间中，四周都是围墙。并且有无数块石头，每个石头都会朝某个方向移动。

首先在一些格子中放上某种类型的石块。然后，可以选择之前放置的其中一块石头并激活它。被激活的石块将会一直朝着它的方向移动，直到撞到了其他石块或者撞到了房间四周的围墙。之后这块石头将停止运动。如果它撞到了围墙，则游戏结束。否则，它所撞击到的石块将被激活，且这一过程将会持续发生。但是，当所有石块被激活的总次数达到10^7后，即使游戏还在进行，游戏也将被强行停止。

倘若石块在撞击到围墙或者其他石块之前至少移动了一步，那么称为一次有效的撞击。

[1]这里下标按对n取模理解，下同。

令$n = 2k$，请布置地图使得有效撞击次数超过$k^3 - k^2$。

规定$n \le 300$。

Analysis

首先考虑连续激发的装置，>.>.，这样激发了最后一个石头之后就能将信号向前传递，进行若干次有效的撞击且整体向前一格。但是这个撞到墙之后前面的东西会停止向前导致长度递减，并且后面的石头向前走之后就不能通过其他方向的信号激发。那么我们还要在后面堆一个储能装置>>>>，这样有两个好处，首先在石头都走完之前一定能通过其他方向的石头激发，然后还能补充激发装置的长度。

接着只要每行放一定长度的激发装置和储能装置，然后串联起来就行了。假设总长度是n，激发装置长度为a，储能装置长度为b，那么撞击次数为$ab/2$，当$a = b = n/2$是取到最大值，于是每行的撞击次数是$k^2/2 - o(k^2)$的，所有撞击次数是$k^3 - o(k^3)$的。

难度1/5。

Comment

这个题是dolphinigle出的，CF上原题是只要构造$n = 100$次数超过10^5的解就行了。这个人在TC上出过很多比较难的题，出这样的题也是挺敢玩的。我觉得他是一个很厉害的出题人，出的题目质量也是挺高的。memsql中也有他的题，不过memsql好像是pieguy压轴，所以我觉得memsql的题目质量也很高，和GCJ有一拼。

关于这个题的理论上界是n^3，因为每个石头朝的方向不会走超过n步，是否能证明理论上界是$n^3/8$，正在思考中，现在我的结果是$n^3/4$。

1.72 331C The Great Julya Calendar

Description

有一个数字n，每次能减去十进制表示下数位里的某个数，求最少步数减到0。

规定$n \le 10^{18}$。

Analysis

令$f(n)$为n的答案，首先证明$f(n)$不降。对n归纳，当$n = 1$时成立，假设当$n \le k$时成立，当$n = k + 1$时，令$d(n)$表示n中最大数位。由归纳假设得$f(n) = f(n - d(n)) + 1$，所以欲证$f(n) \ge f(n-1)$，只需证$f(n - d(n)) \ge f(n-1-d(n-1))$，也就是$n - d(n) \ge n-1-d(n-1)$，即$d(n-1)+1 \ge d(n)$，假设$n = 10q + r$，其中$r$为$n$的个位，若$r = 0$，那么$d(n-1) = 9$显然成立，若$r \ne 0$，那么$d(n-1) + 1 = \max(d(q)+1, r) \ge d(n) = \max(d(q), r)$。证毕。

所以只要对数字n每次减去数位中最大的数，求几次减完就行了。考虑dp，状态为(i, j, k)表示前面的数位最大权为i，考虑后j位，当前后j位为$10^j - k$，其中$1 \le k \le 9$，只要求出把后j位减完的步数，同时由于最后一步不一定恰好减为0，所以还要记录最后一步减到了哪个数。

时间复杂度$O(\log n)$，难度1/5。

1.73 331D Escaping on Beaveractor

Description

二维平面，其中点$x, y (0 \le x, y \le b)$属于区域内部。

区域内部有n个箭头，即为平行于坐标轴的有向线段，箭头之间没有公共点。当行走时遇到箭头方向就会变成箭头所指的方向，每个时刻都会向当前方向走一单位。

有q个询问，在0时刻从(x_i, y_i)出发，向w_i方向移动，问t_i后的位置，或者求出离开区域内部时的位置。

规定$n, b, q \le 10^5, t_i \le 10^{15}$。

Analysis

首先离线使用扫描线加线段树求出每个箭头或询问接下来走到那个箭头或者区域外部。

然后使用倍增即可，求出从箭头i开始走2^j个箭头能到达箭头和步数。

时间复杂度$O(n \log n + \sum \log t_i)$，难度1/5。

Comment

此题虽然很简单，但是代码量不小，只有tourist在比赛中过掉了这个题，不过如果有一个半小时以上的时间写这个题也不是完全没有希望。

1.74 332D Theft of Blueprints

Description

给出一个n个点的带权无向图，满足对于任意一个大小为k的顶点集合S，恰好有一个点与S每一个点都有边。令这个点为$v(S)$，并且对S进行操作的代价是S中每个点与$v(S)$的边权之和。现在求对于一个大小为k的子集操作代价的期望。

规定$n, k \le 2000$。

Analysis

考虑点u有$n (n \ge k)$个相邻点，那么就有$\binom{n}{k}$个子集，每条边出现在$\binom{n-1}{k-1}$个子集中。因为每个k 元子集恰有一个公共点，所以这样是不重不漏的。所以这样做能得到边权的总和和子集的总个数，就能得到期望了。

事实上我们有更好的结果，对于任意$k \ge 3$，此图是一个$k+1$个点的完全图。首先这个图一定是个连通图，若这个图不是连通图，那么可以任选一个k元集合，使它不在同一个连通块内，恰有一个公共点，显然矛盾。又由于$n > k$，任选一个k 元子集，就有一个点与这些点相邻，记为u，那么u 的度数不小于k，记u 的度数为n，假设u 的相邻点集合为S，那么对于任意S 中的$k-1$元子集，并上u形成了一个k元集，那么这k元集有一个相邻点，且这个点与u 相邻，所以在S 中。若有两个$k-1$元子集A, B 与u 的并有一个相同的相邻点v，那么因为$|A \cup B| \ge k$，所以任意k 元子集$C \in A \cup B$

与v相邻，但因为$C \in S$，所以C与v 相邻，矛盾。所以形成了$k-1$ 元子集与S中点的单射，也就是$\binom{n}{k-1} \le n$，即$\binom{n-1}{k-1} * (n-k+1) \le 1$，当$n=k$是等号成立，所以这些点构成了一个$k+1$个点的完全图。又由于这是个连通图，所以就是一个$k+1$个点的完全图。

当$k=1$时，显然是若干个点两两配对连边，当$k=2$时，一定是v_1与$v_2, v_3, \cdots, v_{2k+1}$连边，然后$v_{2i}$和$v_{2i+1}$，其中$i=1,2,\cdots,k$。

Comment

这题应该是数学竞赛原题，虽然不知道出处，但是我从多本书上看到过这个题。

1.75 333C Lucky Tickets

Description

如果数之间加入运算符号和括号使得最终结果k，那么即为k-lucky ticket。

求m个8位的k-lucky ticket，允许有前导0。

规定$0 \le k \le 10^4, m \le 3 * 10^5$。

Analysis

首先将0000到9999之间所有的数字爆搜出通过加减乘除能得到的数，这些得到的数显然不会超过9999。同时也能对每个数处理出哪些数能得到这个数。

然后将前后两部分组合，中间用加或减连接。枚举前面能得到数字x，那么后面4位只要能得到$|k-x|$即可，然后进行一下判重。

难度1/5。

Comment

在当时比赛的时候，我想到了这个方法，但是没有去写，因为题目要求构造$3 * 10^5$个解，而你根本无法知道这个做法的实际效果。

1.76 338D GCD Table

Description

给定一个序列$a_1, a_2, \cdots, a_k$，问是否存在$i, j(1 \le i \le n, 1 \le j \le m-k+1$，使得对任意$\gcd(i, j+l-1) = a_l (1 \le l \le k)$。

规定$n, m \le 10^{12}, k \le 10000$。

Analysis

记$s = \mathrm{lcm}(a_1, a_2, \cdots, a_k)$，那么一定有$s|i$。如果$i$满足条件，那么$s$一定满足条件。考虑每个素数$p$，令$f(x)$表示为$x$中$p$ 的幂次，于是有$\min(f(i), f(j+l-1)) = f(a_l)$。若存在$f(j+l-1) > f(i)$，那么一定有$f(i) = f(a_l) \le f(s) \le f(i)$，所以$f(i) = f(s)$，即$s$ 也满足条件。否则就有对所有l都有$f(j+l-1) \le f(i)$，于是就有对所有l都有$f(j+l-1) = f(a_l) \le$

$f(s)$，所以$\min(f(s),f(j+l-1))=f(a_l)$同样成立。对所有素数都成立，所以$s$一定也满足条件了。

接着考虑j，考虑s的素因子p，其中$p^k||s$，那么必有$p^k|a_l$，于是就有$j+l-1\equiv 0 \pmod{p^k}$。这构成了一个同余方程组，可以使用中国剩余定理解出$j\equiv j_0 \pmod{s}$，然后再暴力判断是否符合条件即可。

时间复杂度$O(k\log n)$，难度1/5。

Comment

此题的主要难点在第一步，即使用最大公倍数代替即可，当然考虑2的幂次的时候就不难得到这个结论。接下来求j的值就简单多了。

1.77 338E Optimize!

Description

给定两个数组$a_1,a_2,\cdots,a_n$和$b_1,b_2,\cdots,b_m$，求有多少个i其中$1\le i\le n-m+1$，$a_i,a_{i+1},\cdots,a_{i+m-1}$这些数与$b_1,b_2,\cdots,b_m$能一一匹配，两个数$a,b$能匹配当且仅当$a+b\ge h$。

规定$m\le n\le 150000$。

Analysis

Solution 1

求b和s的匹配时，考虑贪心算法，对于每个b_i，可以找到一个比$h-b_i$大的最小的s_j，然后从s数组中删掉。如果s_j不存在，那么就无解。

首先如果m很小，那么我们可以暴力做，复杂度是$O(nm)$。

然而$a[i\cdots i+m-1]$和$a[i+1\cdots i+m]$之间的公共部分是$a[i+1\cdots i+m-1]$。如果我们先处理$a[i+1\cdots i+m-1]$，假设s数组剩下的是s'，那么判断$a[i\cdots i+m-1]$和$a[i+1\cdots i+m]$可以看成这样的操作：插入a_i，判断，删除a_i，插入a_{i+m}，判断。

我们选取一个Y，然后同时考虑所有从$[i-Y,i]$开始的数组，这些数组的公共部分是$[i,i-Y+m-1]$，我们可以在$O(m-Y)$的复杂度先解决这一部分，设剩下的数组是s'。接下来我们只要考虑多出来的$O(Y)$个数和s'是否满足条件了，每次都能在$O(Y)$的复杂度解决，所以解决$[i-Y,i]$这一段的复杂度是$O(Y^2+m)$。由于每Y个分一段，总共是$O(n/Y)$段，所以总复杂度是$O(n*Y+m*n/Y)$，Y取到$\sqrt{m}$。

时间复杂度$O(n*\sqrt{m})$，难度1.5/5。

Solution 2

我们把它看成一个二分图匹配，那么重要条件就是Hall定理：对于X中的每个集合S，都有$|N(S)|\ge|S|$。

对于一个数集S，$N(S)=\{a|a\in Y\&\max(S)+a\ge h\}$，即对于每个数在X中的数，满足$\{b|b\in X\&b\le x\}$的大小不大于$\{a|a\in Y\&h-a\le x\}$。

我们把所有数b_i和$h-a_i$放到数轴上，令b_i权值为-1，a_i权值为1，我们只要维护以b_i结尾的前缀和非负，也就是最小前缀和问题，同时要支持+1，-1的操作，这可以用线段树解决。

时间复杂度$O(n\log n)$，难度1/5。

期望得分100分。

Comment

这个其实是一个原题POI 2009的lyk，原题是：

有1到n号的溜冰鞋各k双。已知x号脚的人可以穿x到$x+d$的溜冰鞋。有m次操作，代表来了或走了x_i个r_i号脚的人。对于每次操作，输出溜冰鞋是否足够。

显然POI 2009的题比这个题难一点。作者提供的标算Solution 1还是挺有趣的。

1.78 339E Three Swaps

Description

有一个$1\sim n$的排列，进行三次操作，每次选择一个区间$[l,r]$，将这段区间内的数翻转。

给定最后的排列，求出一个可行的操作序列。

规定$n\leq 1000$。

Analysis

首先如果最终的序列中有4个连续的数字，比如$a,a+1,a+2,a+3$，那么一定存在一种方案不会在过程中分开$a+1,a+2$。如果在某次操作中分开了$a+1,a+2$，那么第三次操作显然不可能，如果第二次操作分开了，那么第三次操作再把它翻转过来，所以第二三次操作是无效的，那么只可能在第一次操作中分开。不妨第一次操作选择了$[l,a+1]$这个区间，那么第二次操作区间不可能包含$a,a+1$这两个数，如果把它翻转回来，那么这两次操作可以抵消，否则$a,a+1$这两个数顺序，不可能在第三次翻转中回到$a+2,a+3$旁。于是第三次操作才把$a,a+1$翻转回来。那么这个过程中第一三次操作不包含$a,a+1$，所得答案是不变的，证毕。

于是因为三次操作最多讲序列分成了7段，每段只可能两端点和两端点向内一个位置可能成为分界点，所以可行的分界点是$O(1)$的，于是可以枚举并适当根据段数加上剪枝。

时间复杂度$O(n)$，难度0.5/5。

Comment

一开始没有证明，觉得只有可能每段的两个端点成为分界点，于是WA掉了。所以进行了一下证明，证明了算法的正确性才会更安心一点。

事实上各种各样的暴力经过一些剪枝还是可以过的。

1.79 341E Candies Game

Description

有n个数$a_1,a_2,\cdots,a_n$，你可以进行若干次操作，每次选择两个数i,j其中$a_i\le a_j$，然后将a_i,a_j变成$2a_i,a_j-a_i$。

问经过不超过10^6次操作能否将使数列中只有2个数非0。

规定$n\le 10^3,\sum_{i=1}^n a_i\le 10^6$。

Analysis

考虑三个数a,b,c，如果能实现一系列操作使得其中恰有一个变为0，那么就能完成目标。不妨设$a\le b\le c$

容易想到要使用辗转相除的手段，考虑$[\frac{b}{a}]$的二进制展开，不妨设为$\sum 2^{k_i}$，其中$k_i>k_{i+1}$。将a,b或a,c进行上述操作就能将a翻倍，如果此时第一个数为$2^x a$，并且x出现在k_i中，那么从b中减去，否则从c中减去，操作结束后b就变成了$b-[\frac{b}{a}]*a=b \bmod a$，同时因为$c$中减去的数字不会超过$(2^{k_1}-1)a<b\le c$，所以第三个数还是大于0，然后不停重复这个过程。

这样做看上去挺正确的，但是辗转相除基于一个数不变，另一个数至少变成一半的事实，从而保证复杂度，然而这样做a,b两个数都变了，无法说明在对数步内就能完成，所以需要补救这个做法。

令$x=[\frac{b}{a}]+1$，考虑x的二进制展开，设为$\sum 2^{k_i}$，其中$k_i>k_{i+1}$。按按上面的过程知道第一个数变成$2^{k_1}a$，第二个数为$b-(x-2^{k_1}*a)$，这个时候用第一个数减第二个数就能得到$xa-b=b-b \bmod a$，此时c中减去的数字不超过$(2^{k_1}-1)a\le(x-1)a\le b$，第三个数字大于等于0，但是第一个数字此时一定不为0。

所以如果$b \bmod a\ge a/2$，用第二种构造，否则用第一种构造，这样能保证最小数变成至多为一半，所以能保证复杂度。

时间复杂度$O(n\log^2 W)$，难度1.5/5，其中$W=\sum_{i=1}^n a_i$。

Comment

二进制和辗转相除都比较容易想到，但是将两者联系起来，还是挺不容易的。

如果只考虑第一种构造，也是能通过所有的数据的，但是我不会证复杂度，并且我相信作业中写了这个复杂度的有大多数人是不会证的。当然这个复杂度有可能是对的，我也相信它是对的，因为如果假设余数随机分布，那么期望步数是$O(\log W)$的。不过我最多只能证明能在$O(n\log^2 W)$的代价做到$O(logW)$ 个数非零，并且能在$O(W\log\log W)$ 的时间复杂度内做完。当然我想有可能能说明这样做是对的，但是我想到了那个构造之后就没有继续想下去。

1.80 342D Xenia and Dominoes

Description

有一个$3*n$的拼图，里面有些方块为障碍，并包含若干骨牌。一个拼图被称作合法当它符合以下条件:

- 每个骨牌正好覆盖两个非障碍方块。

- 没有两个骨牌覆盖同一块区域。
- 有且仅有一个非障碍方块没被任何骨牌覆盖。
- 存在骨牌可移动。

给定$3*n$的拼图和不被任何骨牌覆盖的非障碍方块的位置，问有多少的合法的拼图。

规定$n \le 10^4$，答案对10^9+7取模。

Analysis

考虑枚举能向四个方向移动，然后容斥即可。如果能向左移动，那么一定有一个骨牌覆盖左边两格，于是将这两格和这个不能被覆盖的方块设为障碍，其他同理，然后用状压dp求出有多少种可行的方案即可。

时间复杂度$O(n)$，难度0/5。

1.81 343E Pumping Stations

Description

有个n个点m条边的网络，定义$f(a,b)$表示a到b的最大流，求结点的一个排列$v_1, v_2, \cdots, v_n$，使得$\sum_{i=1}^{n-1} f(v_i, v_{i+1})$ 最大。

规定$n \le 200, m \le 1000$。

Analysis

首先构出Gomory-Hu tree，构造Gomory-Hu tree有n次最大流的做法。由Gomory-Hu tree的性质可得$f(a,b)$为树上a, b 结点之间最小边权。

假设树上的边从大到小排序为$d_1, d_2, \cdots, d_{n-1}$，那么$f(v_i, v_{i+1}) \ge d_k$的对数不会超过$k$个，考虑这$k$条边构成的子图，$v_i$和$v_{i+1}$一定要在同一个连通分量内。又因为每个连通分量都是一棵树，假设由p条边构成，那么有$p+1$个结点，就至多提供p个点对使得$f(v_i, v_{i+1}) \ge d_k$，对所有连通分量求和即得结论。所以显然有$\sum_{i=1}^{n-1} f(v_i, v_{i+1}) \le \sum_{i=1}^{n-1} d_i$。

接着给出构造，首先选取最小边，将树分成两部分，递归构造，然后将两部分的路径端点连起来，正确性归纳易得。

时间复杂度$O(n * \text{maxflow}(n, m))$，难度0.5/5。

Comment

此题的作者在比赛前减小范围让n^2次网络流也能通过，我觉得这个行为是相当的明智的。虽然给人一种拼题之嫌，但是树也能看成网络流问题的一个内在的性质，所以结合得也算不错。

1.82 348E Pilgrims

Description

有一个n个结点的带权的树，其中m个结点中有人，每个人要去离他最远的有人的点，如果有多个，那么这个人会把所有的都列入清单。

要求删去一个没有人的结点，使的尽量多的人在清单中没有任何一个点能去，并求方案。

规定$n \le 10^5$。

Analysis

首先求出这棵树的直径和中心，如果中心在边上，那么可以在边中间加入这个点，记中心为p，半径为r。令p为这棵树的根。

对于每个点u，记dep_u表示到p的距离，$d(u,v) = dep_u + dep_v - dep_{LCA(u,v)} \le dep_u + r$，当$u,v$在$p$的不同子树中，且$v$到$p$的距离为$r$时取到等号。记$w$为$p$子树中除了$u$所在子树其他满足$dep_v = r$的点的LCA，也就是所有可行$v$点的LCA，那么只要删除$w$到$u$路径上任意一点，就能保证$u$清单中的点不能到达。关于$w$可以使用前缀和与后缀和解决，记$f_i$表示前$i$个子树满足条件点的LCA，$g_i$表示后$i$个子树满足条件点的LCA，那么除去第$i$个子树的LCA 就是$f_{i-1}$和$g_{i+1}$的LCA。

然后可以用dp，求出删除每个点能使多少人不满足。

时间复杂度$O(n \log n)$，难度1/5。

1.83 351D Jeff and Removing Periods

Description

有一个数列$a_1, a_2, \cdots, a_n$，可以进行下列操作，选择一个数值相等的等差子序列，即$a_v = a_{v+t} = \cdots = a_{v+tk}$，然后将其删去，将剩下的数重排。

给定数列$b_1, b_2, \cdots, b_m$，有q个询问，问$b_l, b_{l+1}, \cdots, b_r$这段序列需要多少次操作能删完。

规定$m, q \le 10^5$。

Analysis

首先发现如果能一次重排，那么可以将相同的数字排到一起，然后一次删去即可，那么代价就是不同数字的个数。如果第一次操作时没有一种数构成等差子序列，能一次性删除，那么代价就是不同数字个数加一。

所以对每个询问，只要知道不同的数字个数，和有没有一种数的下标构成等差数列即可。对每种数字进行扫描，用two pointer能求出以这个数开始等差数列最长延续到的位置，那么对应一个左右端点分别在某个区间内的询问是满足的，也就是给二维区间打上标记，这个可以只用离线加树状数组解决。

不同的数字是经典问题，移动右边界，将当前位置和当前数前一次出现的位置对应的左边界加一即可。

时间复杂度$O(n \log n)$，难度0.5/5。

Comment

这是sereja的题，他提供的标算是根号划分，也就是莫队算法，也不知道是什么心态。

1.84 360D Levko and Sets

Description

有两个数列$a_1, a_2, \cdots, a_n$和$b_1, b_2, \cdots, b_m$和一个质数p，生成n个集合，第i个集合一开始只有一个元素1，从此集合中任意选出一个元素c，对于所有j，如果$c * a_i^{b_j} \bmod p$不在当前集合中，则把它加入当前集合，重复此过程知道无法将任意元素加入集合。求这n个集合并的大小。

规定$n \leq 10^4, m \leq 10^5, p \leq 10^9$。

Analysis

首先令g为p的原根，做离散对数。令$d_i \equiv \log_g a_i \bmod p$，原过程等价于对集合中任意元素$c$将$(c + d_i * b_j) \bmod (p-1)$加入到集合中。记$f_i = d_i * \gcd(b_1, b_2, \cdots, b_m)$，由一些简单的数论知识可得集合为$\{f_i * j \bmod (p-1) | 0 \leq j \leq p-1\}$，由简单的数论知识可以得到，也就是0到$p-2$之间能被$\gcd(f_i, p-1)$整除的所有数构成的集合。因为我们只要知道$\gcd(f_i, p-1)$，所以只要知道$\gcd(d_i, p-1)$，也就是$a_i$的阶即可，所以只要枚举$p$的约数即可。

接着求这些集合的并，令集合S_d表示0到$p-2$之间能被d整除的数其中$d|p-1$，令T_d表示0到$p-2$之间与$p-1$公约数为d的集合，那么有$S_d = \cup_{d|d'} T_{d'}$，并且这些T集合两两不相交，只要将每个S集合划分成若干个T集合的并，那么所有的S集合的并也能划分成若干个T集合的并。只要求T集合的大小就行了，T集合的大小可以尝试直接使用欧拉函数计算或者只用容斥。

时间复杂度$O(nd\log n + m + d^2)$，其中d为$p-1$的因子个数，难度0.5/5。

Comment

简单的数论知识指裴蜀定理和由数学归纳法可得的一些推论。这个题看上去比较麻烦，但实际上只要知道原根，接下来的一些处理手法还是挺显然的。

2 USACO

2.1 Dec 05 Cow Patterns

Description

有两个数组$a_1, a_2, \cdots, a_n$，$b_1, b_2, \cdots, b_m$，其中满足$1 \le b_i \le S$，问有多少$i(1 \le i \le n-m+1)$，满足$a_i, a_{i+1}, \cdots, a_{i+m-1}$与$b_1, b_2, \cdots, b_m$匹配，这里匹配指数字大小关系相同。

规定$n \le 10^5, m \le 25000, S \le 25$。

Analysis

如果两个数字匹配，那么它们的前缀也是匹配的，所以可以使用KMP算法。那么要判断的是两个前缀同时加入一个数字大小关系是否相同，一个直接的想法是使用可持久化数据结构维护排名，但是这样比较麻烦。于是对于每个数，维护一下前缀中比它大的最小的数和比它小的最大的数和相等的数，这样就能比较了。

时间复杂度$O(n \log n)$，难度0.5/5。

Comment

此题标算看上去是和S有关的。这个题还是CEOI 2011 Matching，这里使用的做法就是CEOI这个题的做法。

2.2 Jan 07 Cow School

Description

有n个点(x_i, y_i)，一个点集的斜率定义为所有点y坐标和与所有点x坐标和的比值。

问有哪些d，存在大小为$n-d$的点集的斜率超过n个点中去掉y_i/x_i最小d个产生的点集。

规定$n \le 50000$，没有两个点到原点斜率相同。

Analysis

如果单单询问大小为d的斜率最大的点集，可以二分答案判定，但是这样无法高效的维护。

然而问题是给定点集，判定是否最大。假设当前点集的斜率为k，若不是最大，也就是存在大小为d的点集，其中点的$y_i - k * x_i$加起来超过0，那么只要判断点集中$y_i - k * x_i$最小的是否小于不在点集中$y_i - k * x_i$最大的值，如果小于，那么可以做出调整得到更大的点集。

所以只要枚举d，支持插入一个点，查询点集中$y_i - k * x_i$最大的点。删除一个点可以倒序操作变成插入一个点。这个可以使用平衡树进行维护。

时间复杂度$O(n \log n)$，难度1/5。

Comment

这个题在当时那个年代应该算挺难的了，当然我觉得即使在现在也不是太简单。当然这个题可以使用对时间分治的技巧达到同样的复杂度，并可

以避免动态凸包，不过对于有模板的人来说也无所谓了。

题解中提到了一个线性的做法，不过网上也有说法那个是错的，我也没有仔细查看。

2.3 Open 07 Connect

Description

有一个$r*c$的网格图，支持m个操作，相邻两点之间连边删边，询问(r_1,c_1)和(r_2,c_2)两点在纵坐标在c_1和c_2之间的点的导出子图中是否连通。

规定$1 \le r \le 2, c \le 15000, m \le 50000$，要求在线。

Analysis

用线段树维护$[l,r]$这段区间内四个端点之间的连通性，合并时候只要做一遍floyd即可。

时间复杂度$O(m\log c)$，难度0/5。

Comment

这个题在Tsinsen上版本使用加密的方式实现在线，不幸的是出题人将r也异或上了$key_{lastans}$，而key是随机生成的，因为$1 \le r \le 2$，所以基本可以确定lastans，所以只要解密就能知道答案了。但是这样如果最后一问是询问还是不知道答案的，不过有趣的是这种情况下数据中答案都是0。

这样的题目还有SHOI 2008 堵塞的交通，维护$2*n$网格的连通性，WC 2009 shortest，维护$6*n$网格图最短路。

更早的是WC 2005双面棋盘，在那个科技没有现在发达的年代出现这样的题还是挺令我震惊的。

2.4 Dec 07 Best Cow Line

Description

有一个字符串s，每次选择从头或尾删除一个字母，并加入新的序列中，求字典序最小的方案。

规定$|s| \le 30000$。

Analysis

考虑贪心，记s^R为s反转后的串，如果s字典序小于s^R，那么删头，否则删尾。若$s=s^R$，那么显然等价，否则假设a,b为s和s^R出现不同的第一个字母，取到a或b之前s和s^R等价，那么显然选择a,b两字母中小的一个，于是贪心成立。

要判断这个只要使用二分+hash即可。

时间复杂度$O(|s|\log|s|)$，难度0.5/5。

Comment

我怀疑这个题存在不使用后缀数据结构的线性的做法。

2.5 Mar 08 Land Acquisition

Description

有n个点构成的点集，每次可以选择一个子集删去，代价为子集中点横纵坐标最大值的乘积，求最小的代价将点全部删去。

规定$n \leq 50000$。

Analysis

首先如果存在点i, j满足$x_i \leq x_j$且$y_i \leq y_j$，那么i和j一起删去不会更差，同时i并不会对答案造成影响，所以可以把这样的i都删去，剩下的m个点满足$x_1 < x_2 < \cdots < x_m$且$y_1 > y_2 > \cdots > y_m$。

如果一次删去的点集编号最小和最大的分别为l, r，那么将l, r之间点一起删去不会更差。

令dp_i表示删到第i个点最小代价，于是有$dp_i = \min_{j=1}^{i-1}(dp_j + y_{j+1} * x_i)$，这显然可以使用斜率优化，并且$x$和$y$有单调性，使用队列维护即可。

时间复杂度$O(n \log n)$，难度0/5。

Comment

此题为经典文1d1d动态规划优化中的例题，充满了教育意义。

2.6 Open 08 Cow Neighborhoods

Description

有n个点，将曼哈顿距离不超过C的点之间连边，问有多少个连通块及最大连通块大小。

规定$n \leq 10^5$。

Analysis

首先旋转坐标轴45度，问题就变成了向以这个点为中心$2C * 2C$的正方形连边。

每个点只考虑下面的点，也就是每个点向左下方$C * C$和右下方$C * C$的正方形内的点连边。每个点只要向这两个正方形[2]里面y 坐标最大的点连边即可。因为剩下的点都在这两个y坐标最大点的矩形范围内。y坐标相同的点直接特判即可。对y坐标归纳可证其正确性。

实现的时候，可以从小到大枚举y坐标，使用线段树维护纵坐标在当前y坐标到当前y坐标减C的点即可。

时间复杂度$O(n \log n)$，难度0.5/5。

Comment

此题出现在ONTAK2010，这个好像是波兰高中生的一个比赛，不过似乎是练习性质的，所以出现了挺多的原题。

[2]其中不包含上边界。

2.7 Jan 09 Safe Travel

Description

有n个点m条边的带权有向图，每个点到1的最短路唯一。求对每个点i删去i到1最短路上第一条边后的最短路径。

规定$n \le 10^5, m \le 2 * 10^5$。

Analysis

首先建出最短路树，对于每个点，一旦走上一条非树边后，那么就避开了删去的路径，也就是一定会沿着树边走上最短路，并且由于要避开删掉的边，那么路径上一定有一条非树边。

于是可以枚举非树边，计算对答案的贡献。假设u点的最短路是dis_u，一条非树边为(v, w)，那么u点一定要在v到w的路径上，且不为v和w的最近公共祖先，否则无法通过树边到达v这个点或者无法绕过被删掉的边，记这条边边权为c，那么新的路径权值为$dis_v + c + dis_w - dis_u$，将$dis_u$ 提取出，可就是用$dis_v + c + dis_w$去更新可行的u点。

这个可以使用树链剖分或者可并堆等数据结构实现。

时间复杂度$O((n+m)\log n)$，难度0.5/5。

Comment

这题是关于删边最短路中比较经典的一个题，但是在此题之前就已经出现过k短路的题了。

2.8 Mar 09 Cleaning Up

Description

有一个数列$a_1, a_2, \cdots, a_n$，求把数列划分成若干段，每段的权值为其中不同数字的个数的平方，求最小的总权值。

Analysis

首先每段中不同数字个数不会超过$\sqrt{n}$，否则将每个数字单独划分成一段不会更劣。所以只要维护到每个位置最近出现的$O(\sqrt{n})$个数然后进行转移即可。但是无法瓶颈是转移似乎无法优化。

时间复杂度$O(n^{1.5})$，难度0/5。

Comment

此题还是2014年西安赛区网络赛C题，由复旦大学命题，也不知道是怎么回事。

2.9 Mar 10 StarCowCraft

Description

有三个正实数变量x, y, z，并且没有一个值是另一个100倍以上。

有n个$a_i * x + b_i * y + c_i * z \le 0$或$a_i * x + b_i * y + c_i * z \ge 0$的限制。

有m个询问，问能否判断$a_i * x + b_i * y + c_i * z$大于0或小于0。

Analysis

令$x' = x/z, y' = y/z$，那么有$0.01 \le x', y' \le 100$，且$0.01x \le y \le 100x$，同时每个限制对应一个半平面，只要做半平面交即可。

对于每个询问，只要判断交集是否在某个半平面的一边，只要判断每个顶点即可。

时间复杂度$O(nm)$，难度0.5/5。

2.10 Open 10 Triangle Counting

Description

给定n个点，问有多少三角形包含原点。

规定$n \le 10^5$，这些点包括原点没有三点共线，

Analysis

考虑哪些三角形不包含原点，那么这些点一定在某条过原点的直线的某一边，枚举极角序最小的点，即角度为α，那么剩下的两个点的极角一定在$[\alpha, \min(\alpha + \pi, 2\pi)]$之内，只要排序后用two pointer维护即可。

时间复杂度$O(n)$，难度0/5。

Comment

类似的可以做若干个点的凸包包含原点。如果有三点共线，那么只要将三点共线，同时通过原点的情况去除即可。

这题似乎还是最近的SRM 641的div 1 level 1题，感觉出题人和验题人都忘吃药了。

2.11 Dec 10 Threatening Letter

Description

给定串s和t，求将s划分成最少的份数，使得每份都是t的子串。

规定$|s|, |t| \le 50000$。

Analysis

令dp_i表示划分到i末尾最少的份数，只要知道能从前面那些状态中转移而来，也就是求出i结尾最长子串使得它为t的子串即可。这是后缀自动机的基本操作。同时由于显然的单调性，可以使用单调队列优化。

时间复杂度$O(|s| + |t||\Sigma|)$，其中Σ表示字符集，难度0.5/5。

2.12 Mar 12 Cows in a Skyscraper

Description

有n个数构成的集合，划分成若干个子集，使得每个子集中数字和都不超过W，求最少子集个数。

规定$n \le 18$。

Analysis

考虑状压dp，对每个状态S求出最少的集合个数以及这个集合当前最小权值和，考虑一个新的元素加入，如果这个集合权值超过W，那么要新的一个集合，否则将权值和加上这个数。

时间复杂度$O(n2^n)$，难度1/5。

Comment

这个题还出现在PA 2014 2A Pakowanie，只不过改成了集合和分别不超过$w_1, w_2, \cdots, w_m$而已。事实上我并不会做这个题，不过我不会做这个题还是令我挺悲伤的。

这个题搜索能得到更快的速度。

2.13 Dec 12 First!

Description

给定n个有小写英文构成的字符串$s_1, s_2, \cdots, s_n$，问有多少个串能在改变字符大小顺序后变成字典序最小。

规定$n \leq 30000, \sum_{i=1}^{n} |s_i| \leq 3 * 10^5$。

Analysis

首先将所有串建成字典树，dfs整棵树，进入一个儿子结点表示这个儿子结点在字典序中最小，也就是产生了限制这个子结点对应的字母的序小于其他结点，到达某个串尾，将所有的限制建成图，判断是否有环即可。注意某个串是另一个串的前缀这种特殊情况。

时间复杂度$O(n|\Sigma|^2 + \sum_{i=1}^{n} |s_i||\Sigma|)$，其中$\Sigma$表示字符集，难度0/5。

Comment

这个过程可以使用位运算优化。

2.14 Dec 12 Gangs of Instanbull

Description

有n头牛共m种，有一个牧场，每个时刻有一头牛进入牧场中，如果牧场中没有牛，那么这种牛占据牧场，如果牧场中有牛且种类不同，那么牧场中的牛个数减1，这头牛也不会进入牧场。问最后第1种牛在牧场中最多有多少个，并求字典序最小的方案。

规定$n \leq 10^6$。

Analysis

首先一个自然的想法就是把除了1以外的牛先加入牧场中互相消耗得到一个最小值，然后再将1加入到牧场中。容易证明这个想法是对的。

对于另外的一些牛，如果总和是偶数并且没有一个种群超过总和的一半，那么一定是可以全部消耗完的，可以使用数学归纳法证明，否则最小的答案是最大的种群减去其他种群的和，奇数同理，没有一个种群超过总和的一半，那么最后只剩1个。

于是得到了最后第1种牛在牧场中最多有多少个。如果种群只有两个那么可以将1先放入，再放入2，否则最后能剩下的牛一定等全部都消耗完之后放入。所以只要知道将剩下的第1种牛和其他牛一起消耗完字典序最小的方案就行了。

构造方案的时候只要判断一下是否将要超过一半的种群，如果有就先选这个，否则就选择最小的，只要合理维护一下就能做到线性。

时间复杂度$O(n)$，难度0.5/5。

2.15 Mar 13 Hill walk

Description

有n条线段，每条从(x_1, y_1)到(x_2, y_2)，且$x_1 < x_2, y_1 < y_2$。任意两条线段没有公共点。

从第一条线段开始沿着线段往右走，到边界后掉下来落到正下方的线段上接着走，直到正下方没有线段为止。

问经过多少条线段。

规定$n \le 10^5$。

Analysis

扫描线，用平衡树维护线段上下关系，然后模拟。

时间复杂度$O(n \log n)$，难度0/5。

2.16 Open 13 Photo

Description

有n个变量$x_1, x_2, \cdots, x_n$取0或1，有m条方程，第i条为$\sum_{k=l_i}^{r_i} x_k = 1$，求$\sum_{i=1}^{n} x_i$最大值，或说明无解。

规定$n \le 2 * 10^5, m \le 10^5$。

Analysis

我们考虑如何使用dp解决这个问题。从左到右确定每个变量是否为1。那么dp_i表示考虑了$1 \cdots i$这些变量，并且$x_i = 1$的最大和。接着考虑转移，显然如果i这个变量为1，那么我们要考虑前一个1在哪里。如果i这个变量为1，那么所有覆盖了i这个变量的限制对应的变量都为0。同时如果上一个1是x_j，但是存在方程$[l, r]$满足$j < l \le r < i$，那么$[l, r]$这个区间中间变量和为0，矛盾。所以得dp方程$dp_i = \max(dp_j | \max(l_k | r_k < i) \le j < \min(l_k | r_k \ge i) \& j < i) + 1$

容易发现$\max(l_k | r_k < i)$这一部分是简单的递推。这是个区间询问，可以使用线段树或其他数据结构简单的解决。然后区间的左右边界显然单调不降，可以使用单调队列优化。

时间复杂度$O(n)$，难度0.5/5。

2.17 Open 13 Figure Eight

Description

有一个$n * n$的网格，有些格子有障碍。要在里面放一个8，一个有效的8定义如下：

- 由上下两个矩形构成，并且每个矩形内部至少有一个格子。
- 上面矩形底边为下面矩形顶边的子集。
- 矩形边界上没有障碍。

得分为上下矩形面积的乘积，求最大得分。

规定$n \leq 300$。

Analysis

枚举上面矩形底边对应的区间，从上往下扫，能维护出两端能向上延伸多少，所以只要知道在能延伸到的范围最远的顶边能放在那里，这个使用队列就行了。于是就能得到以某条线段为底边的最大矩形，同理能得到某条线段为顶边的最大矩形。

接下来合并两部分的答案，也就是在底边属于某条线段的矩形面积最大值，这个可以使用类似前缀和的方法实现。

时间复杂度$O(n^3)$，难度0.5/5。