ASC问题选讲

范裕达

入选标准

ASC全称Andrew Stankevich Contest，是俄罗斯一个颇具盛名的传统ICPC比赛，于每年的冬夏各举行一次，以题目难度高、富有新意著称。

我挑选的题目的标准：

1、难度适中，不落俗套

2、我在赛场上AC，长度不超过3k

ASC47 B Borderless

一个长度为n的字符串S，如果S[0,i - 1] == S[n – i, n - 1], 那么T=S[0, i - 1]被称为S的一个border，其中$0<i<n$。

例如aaaaa，那么a, aa, aaa, aaaa 都可以被认为是他的border，一个没有border的字符串被称为Borderless的，例如aaaab。

对于长度为n的字符串，求其字典序第k小的Borderless的字符串。

$0<n<64$, $k<2^{64}$

ASC47 B Borderless

分析其性质

1、一个字符串有Border，代表fail[n - 1] > 0

2、一个字符串有Border，则它必然有长度不超过一半的border

例如字符串abaabab, ab和abaab都是其border。

ASC47 B Borderless

采用逐位确定的办法，核心在于计算以某个字符串为前缀的Borderless的字符串有多少个。

如果不会复杂的问题，请考虑条件弱化的版本，如果没有前缀的约束怎么做。

dp[i]表示长度为i的Borderless的字符串的个数。

枚举最小的Border的长度为j, j * 2 <= i，dp[i] -= \sum dp[j] * 2^(i – 2 * j)

ASC47 B Borderless

如果有前缀的约束，有什么变化？假设约束长度为k。

1、i <= k，dp[i] = 0, 1

2、i > k > j, dp[i] -= dp[j] * 2^(i – j - k)

3、i > j >= k, dp[i] -= dp[j] * 2^(i – 2 * j)

此外，需要判断长度为j的border是否可能存在。

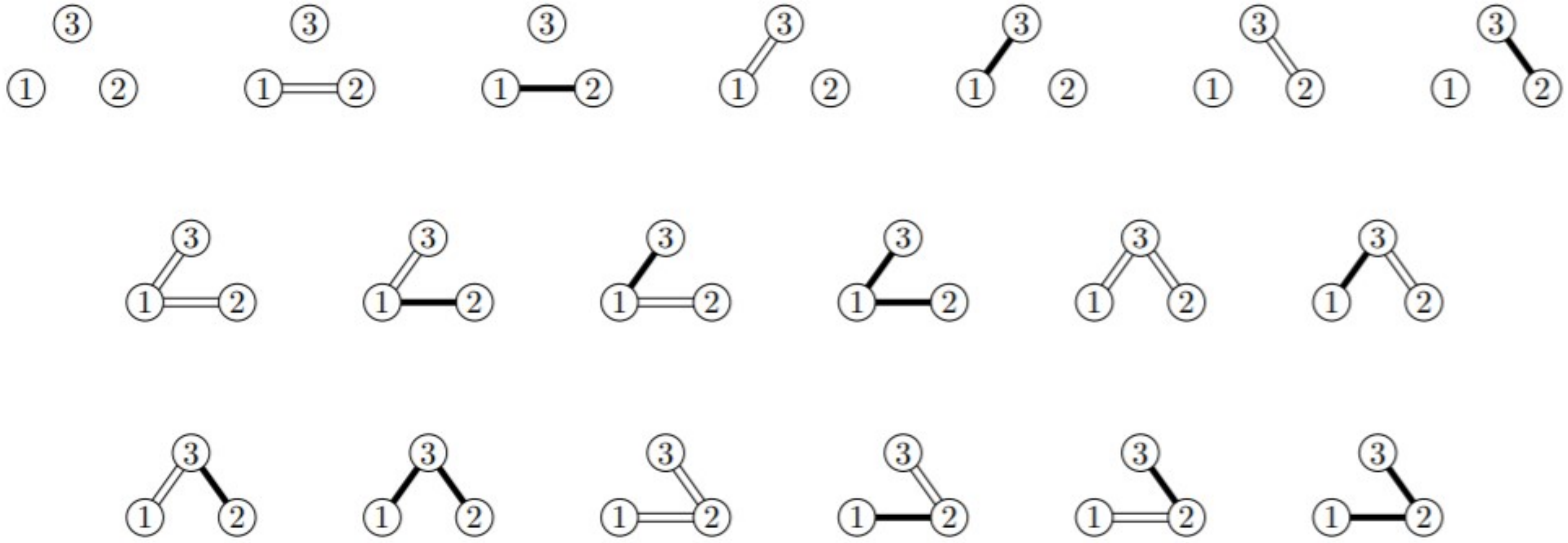
ASC46 B Bipartite Bicolored Graph

对于所有n个点的带标号的二分图，对他的边进行染色，求所有可能染色后的图的个数。

$n < 1000$, 答案取模

ASC46 B Bipartite Bicolored Graph

例子 n = 3 时，共有19种情况



ASC46 B Bipartite Bicolored Graph

换言之，答案就是求

$\sum_G 3^{E(G)}$, G是所有不同的带标号的二分图，E(G)是G边的个数

朴素而错误的想法

枚举X点集大小为i, Y点集大小为n-i，求和。

ASC46 B Bipartite Bicolored Graph

按照连通块进行dp，g[i]表示大小为i的连通图的染色方案个数。

每次新增一个连通块，都规定标号最小的那个点在左边。

f[i] = \sum_j f[i – j] * C(i – 1, j - 1) * g[j], j <= i

g[i] 怎么求?

令 h[i] = \sum_j C(i – 1, j - 1) * 3^(j * (i - j)), j <= i

g[i] = h[i] - \sum_j 2 * C(i – 1, j - 1) * g[j] * h[i - j], j < i

ASC46 B Bipartite Bicolored Graph

如果有前缀的约束，有什么变化？假设约束长度为k。

1、i <= k，dp[i] = 0, 1

2、i > k > j, dp[i] -= dp[j] * 2^(i – j - k)

3、i > j >= k, dp[i] -= dp[j] * 2^(i – 2 * j)

此外，需要判断长度为j的border是否可能存在。

ASC46 E Ebola Virus

n个村庄坐落在一条大路上的村庄感染了病毒，初始第i个村庄感染人数a_i人，
如果不得到救治，每一天这a_i个人都会死去并且传染另外a_i个人。

初始在1号村庄，每天都可以进行两种行动之一:
1、治愈当前所在村庄的所有病人
2、从i号村庄移动到i-1或者i+1

此外，另有一条行动限制，如果你曾今路过某个村庄而没有选择治愈他们，那么
村民就会对你翘首期盼，一旦你朝这个方向再次行进时，你就不能改变方向，直到
你第二次路过它，并且会被强制要求将他们治愈。

求最少死亡人数。

ASC46 E Ebola Virus

考虑行动轨迹

假如从1开始，走到i，最小的没有被治愈的是j，那么如果决定回头的话，则需要一直走到j，并且沿途一一治愈第一次走过遗漏的村庄。

答案是由若干段不相交区间拼成的。对于一个区间[l, r]，只有两种结局：

1、从l走到r，依次治愈所有村庄

2、一开始不治愈l, 先走到r，回头再治愈l，最后回到r

ASC46 E Ebola Virus

宜用前缀和DP

f[i]表示解决完前i个村庄，停留在i的最小牺牲，g[i, j]表示把i到j全部解决，停留在j的最小牺牲,s[i]表示a[i]的前缀和

g[i, j] = g[i + 1, j] + min((s[r] – s[l]) * 2, a[i] * (r - l) * 3)

f[j] = min(f[i - 1] + g[i, j])

ASC45 H Hide And Seek

平面上有一个n个点顺次连接而成的凹多边形房子，小朋友们在里面玩捉迷藏。

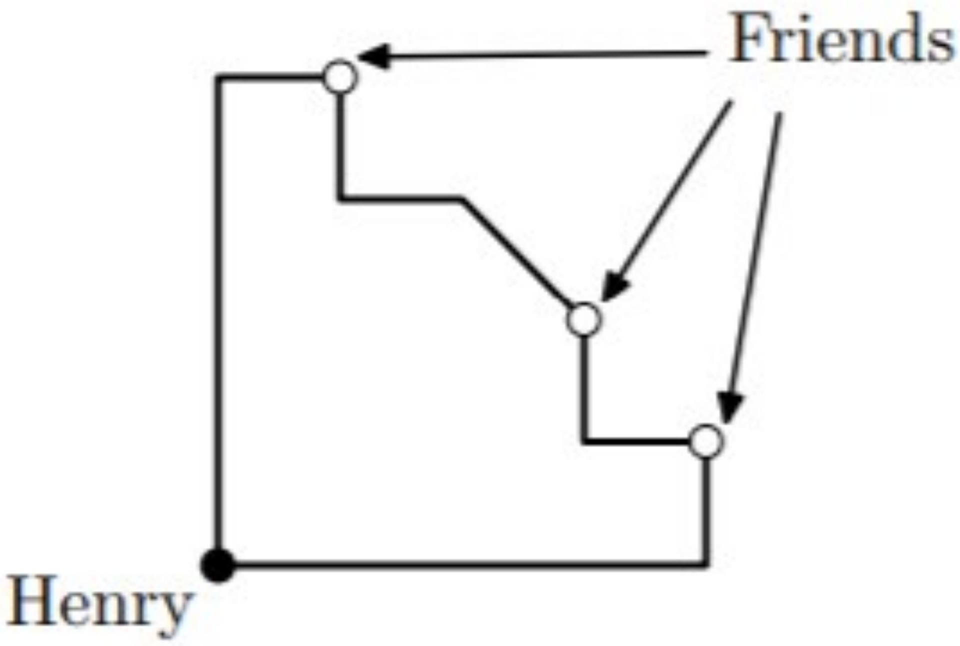
其中1号点由主人Henry占据，其它小朋友则站在各个不同的墙角上。

要求除了Henry，其它小朋友都互相不能看见，问至多有多少个小朋友参加游戏。

$n < 300$

ASC45 H Hide And Seek

一个例子，可以有3个其它小朋友



ASC45 H Hide And Seek

Q1 如何判断两个点是否能互相看见

充要条件

1、两点之间的连线和多边形没有端点之外的交点

2、两点之间的连线处在多边形内

ASC45 H Hide And Seek

分治思想，使用区间DP解决，将答案分解为不可见的互相不可见的区间

$dp[i, j]$表示i到j之中最多选取多少点

显然的$dp[i, j] = max(dp[i, j - 1], dp[i + 1, j])$

若i和j互相不可见，那么找到最小的$i < k < j$

使得k和j互相可见，那么显然有$[i, k - 1]$和$[k + 1, j]$两段区间互相不可见

$dp[i, j] = max(dp[i, k - 1] + dp[k + 1, j], dp[i, j])$

ASC44 F Funny Card Game

长度为n的数列，第i个数为a_i,将其分为k个非空的子序列，子序列的下标都是连续的。每一个子序列的得分是其中众数出现的次数，总得分是所有序列得分之和。

求划分的方案，使得总得分最高。

$n < 1e5$, $k < 100$

ASC44 F Funny Card Game

思考答案性质

在答案中，区间只会有两类

1、长度为1
2、首尾数字是相同的，并且该数字就是这个区间的众数

此外，至多有一个区间不满足以上任意条件。

ASC44 F Funny Card Game

F[i, j]表示前i个数字，分成j段的最高得分。

对于第i个数字，有两种选择：
1、单独成段 f[i, j] = f[i – 1, j - 1] + 1
2、和前面合成一段f[i, j] = f[k, j - 1] + cnt[i] – cnt[k], 其中a[k + 1] == a[i]

对于每个a[i]， f[k, j -1] – cnt[k] 都可以分离出来，具有决策单调性，可以维护其最有决策点

时间复杂度
O(nk)

ASC44 H Huffleman Code

哈夫曼编码是一种最有前缀编码，其性质为：

1、编码互相之间不为前缀关系

2、$\sum_i c_i*p_i$ 最小，其中c_i为编码长度,p_i为出现频率。

现在对n个字母进行编码，要求第i个编码中有u_i个0，v_i个1，询问是否有合法的构造。

$n < 100$

ASC44 H Huffleman Code

例子

3	Yes
1 0	0
0 2	11
1 1	10

例子

2	No
1 1	
1 1	

例子

7	Yes
1 1	10
2 1	001
3 0	000
1 1	01
1 2	110
1 3	1110
0 4	1111

ASC44 H Huffleman Code

最有编码的性质

1、编码节点都是叶子节点
2、所有节点均有两个儿子

从二叉树的性质考虑

X的兄弟是X^1,也就是说只差一个末位1
(x, y)可能的兄弟是(x – 1, y + 1), (x + 1, y - 1)

ASC44 H Huffleman Code

解法一
考虑构造这棵哈夫曼树，用dfs递归处理

对于每个节点，两种情况

1、叶子节点，有符合条件的编码，直接构造
2、内部节点，左右儿子都要存在，且两侧都有合法解，递归构造

解法二
直接构造，事实上对于这些编码来说，总有一个的配对方式是确定的，就是深度最深且零最多的那个。

ASC43 L Least Common Ancestor

定义一个满二叉树之间的父子关系

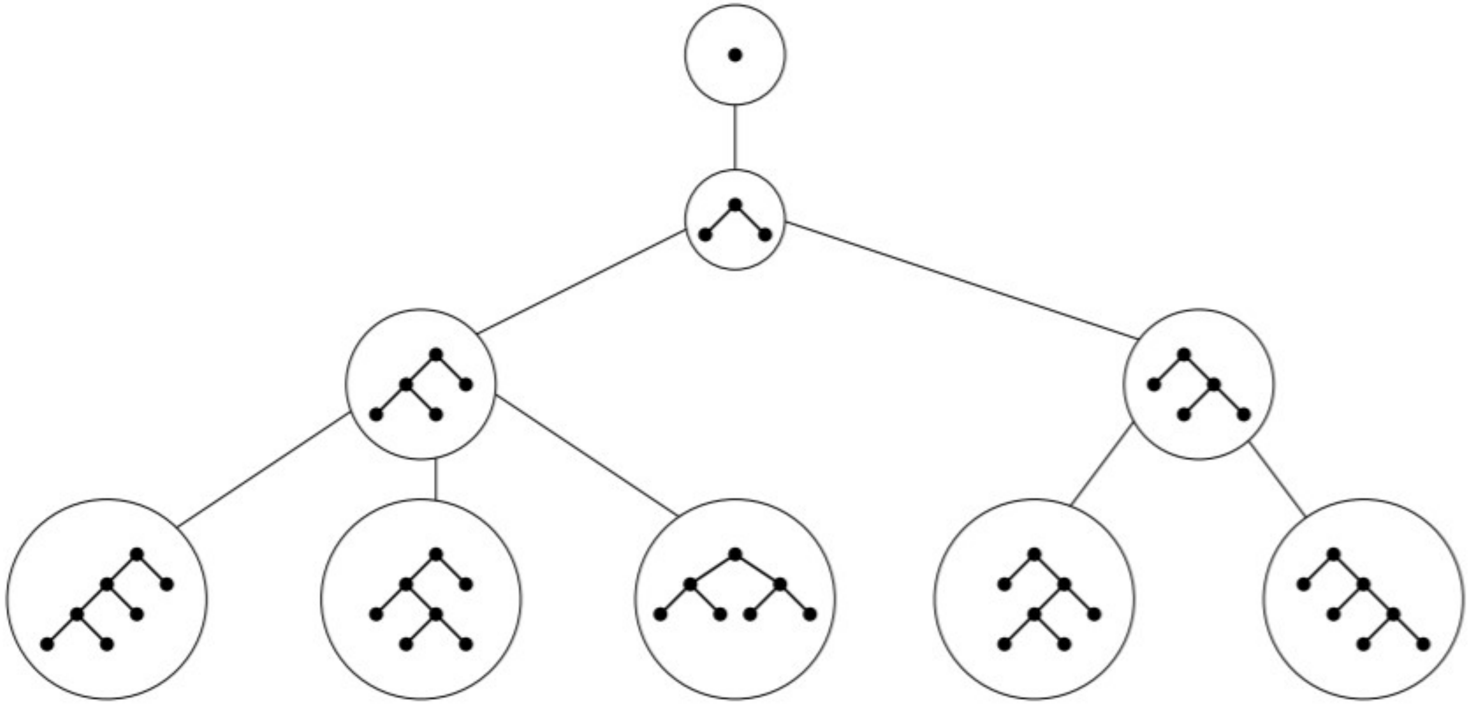
一个满二叉树T，找到其中序遍历最靠后的，且两个儿子都是叶子节点的内部节点x,将其的儿子们都删去，得到其父亲T’。

先给出两棵大小不超过n的满二叉树T1、T2，求在以上关系定义下的两棵树的LCA。

$n < 1e5$

ASC43 L Least Common Ancestor

例子



ASC43 L Least Common Ancestor

考虑Common Ancestor的定义，就是在树结构中，T1和T2都处于它的同一个儿子的子树中，除了Least Common Ancestor，T1和T2分居两侧。

考虑这棵树的形成过程，从父亲到儿子，就是选择一个叶子节点，让其生长出两个新的儿子出来。

如何选择合法的叶子节点呢?

ASC43 L Least Common Ancestor

同时在两棵树上一起dfs，直接构造LCA

对于某个LCA上的节点x，有两种情况：
1、若在两棵树上都有儿子则为内部节点，递归构造
2、反之，则为叶子节点

此外，若x为叶子节点，则其它dfs序在x之后的均为叶子节点。

ASC42 C Comparator Networks

一个排序网络是一系列排序器的序列，每个排序器会负责比较两个位置上的数，并在必要的情况下进行交换。

01法则：对于一个n个数的排序网络，如果所有可能的01序列都能成功被排序，则这个排序网络可以对任何序列成功排序。

先要求构造一个失败的排序网络，它可以对除去某个特定的01序列之外的所有01序列都能成功排序。

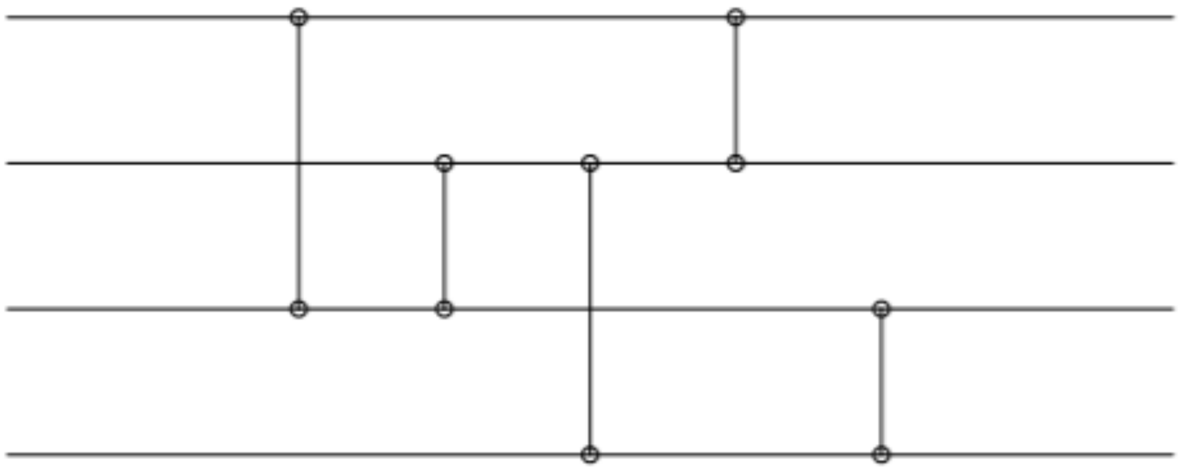
$n<10$
要求构造的排序网络大小不超过1000

ASC42 C Comparator Networks

例子

n = 4

1 0 1 0



ASC42 C Comparator Networks

先说结论

除非给定的序列开始就是有序的，否则一定可以构造出来

再考虑答案形态

一定是解决完了其它所有的逆序对，最后只剩中间唯一的一个，比如 000000101111111

ASC42 C Comparator Networks

我的构造，依次

1、对于0，1集合内部，选择排序

2、抛开第一个1所在的位置，选择排序

3、抛开最后一个0所在的位置，选择排序

你的解法?

ASC42 J Journey Planner

对于一张n个点的有向无环图，找出一个后继结点最多的节点。

不要求最优解，近似比小于2的近似解即可，即你给出的答案必须大于最优解的一半。

$n < 1e5$

ASC42 J Journey Planner

众所周知，如果要求精确解，就得使用bitset压位。

不能直接求解后继结点个数的原因是当有交集时，不能使用简单的加法。

请考虑一下有交集情况下依然成立的运算。

ASC42 J Journey Planner

给每个节点随机[0, 1]内的实数，通过拓扑排序求得所有后继节点的最大值。

n个[0, 1]之内独立均匀分布的随机变量，最大值的期望为n/(n + 1)，可以通过最大值的期望值来估计随机变量个数。

多随机几次即可。

ASC41 E Eating Chocolate

0到正无穷号玩家吃巧克力，共有三块巧克力，大小分别为a, b, c。

从1号玩家开始行动，每轮行动有三步:

1、i号玩家选择一块巧克力保护起来

2、i-1号玩家选择一块巧克力吃掉，这块巧克力没有被保护

3、i号玩家将某块剩下的巧克力一分为二

如果两块巧克力不能再分了（大小都为1了），游戏结束

每个人都是贪婪的，他们既希望吃到最多的巧克力，又希望自己行动的字典序尽量小，即保护更小的巧克力，将巧克力分的尽量小。但是每个人并不知道其他人的策略。

ASC41 E Eating Chocolate

例子
三块巧克力的大小为8, 2, 1

Player	Available pieces	Secured piece	Eaten piece	Division
1	8, 2, 1	8	2	$8 \to 4, 4$
2	4, 4, 1	1	4	$4 \to 2, 2$
3	2, 2, 1	1	2	$2 \to 1, 1$
4	1, 1, 1	1	1	game over

ASC41 E Eating Chocolate

注意要点

每个人并不知道其它人的策略，所以他的想法是使得划分之后次大值最大。

同样的，对于前面一个人，他至少能吃到次大值，因此保护最小的巧克力
没有太大意义，只需分类考虑最大值或者次大值被吃掉的情况。

ASC41 E Eating Chocolate

考虑剩下的两块巧克力，设为A >= B

若A >= 2 * B, 则将A对半分

若 A < B * 2, 则将A分为1和A-1

模拟即可

ASC40 A Average Convex Hull

平面上有n个点，坐标分别为p_i，随机等概率的去掉其中一个点，求凸包上点的个数的期望。

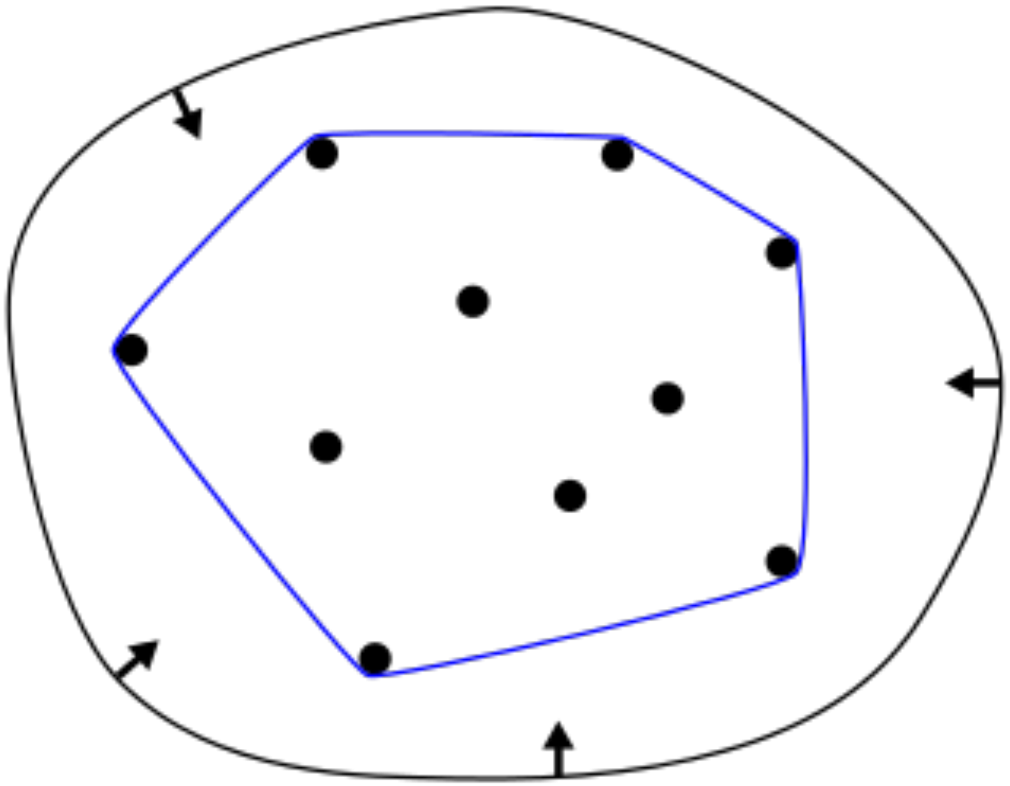
凸包是严格凸包，可认为任意三点不共线。如果凸包退化为线段，则只包括其两个端点。

$n < 1e5$

ASC40 A Average Convex Hull

水平序凸包算法

将点按横坐标排序，从左到右求解下凸壳，从右至左求解上凸壳，进行合并。



ASC40 A Average Convex Hull

考虑被去掉的点的分类

1、在凸包内部，则对于凸包没有影响。

2、在凸包上，则会导致凸包上先失去一个点，之后凸包内缩，边界上可能出现新的若干个顶点。

如何对他们进行统计呢?

ASC40 A Average Convex Hull

这里我们用凸包的顺序来对点编号，考虑去掉顶点p_i

那么如果凸包上出现了一些新的点$x_0, x_1, ..., x_m$，那么这些点必然落在p_i, p_{i-1},p_{i+1}形成的三角形内。

那么显然，三角形p1p2p3和三角形p3p4p5是不相交的，也就是说，顶点2的贡献可以和定点4的贡献一起计算。

ASC40 A Average Convex Hull

推广之，所有奇偶性相同点可以同时计算。

将p_1, p_3, p_5, ... 去掉，计算答案

将p_2, p_4, p_6, ... 去掉，计算答案

在和凸包内部点的答案进行求和

ASC39 H Highways

n个点m条边的无向图，判定是否存在一个导出子图是过于稠密的。

稠密的定义，若对于一个导出子图，其中有n个点，m条边，且满足
$m > k(n-1)$, 则认为该子图是稠密的。

$n < 100$, $m < 5000$, $k < 20$

ASC39 H Highways

先回忆最大密度子图做法

通过分数规划，将问题转化成判定问题，是否存在一个子图密度大于 k。

将边权赋值为1, 点权赋值为-k, 二次转化为最大权闭合子图问题。

利用最小割求解。

ASC39 H Highways

改造判定式

$m > k(n - 1)$, $m > kn – k$

这里多了一个余项-k，我们可以认为是图中的某个点的权值被减去了k。

枚举这个特殊点，改变它的权值。

复杂度O(n^2m)

ASC38 H High School Duels

n个选手参加比赛，比赛的赛制如下：

1、选手们两两之间进行若干场比赛，每场比赛获胜者积一分。
2、赛程结束后积分最高的为胜者，若有同分则并列。

现在赛程已经进行了一段时间了，已知当前每个选手的得分为s_i，
选手i和选手j之间还要进行$a_{i, j}$场比赛。

求问每个选手是否还有成为胜者的可能。

$n < 30$, $a_{i, j} < 1000$, $s_i < 1000$

ASC38 H High School Duels

考虑钦定第i个人成为优胜者

1、显然的，第i个人所有剩下的比赛都要赢。
2、其它的n-1个人互相之间要配合默契，每个人的胜场都不超过i的最大胜场。

ASC38 H High School Duels

用网络流做可行性判定

选手i和选手j对战a_{i,j}，等于将这个多积分分配给二人。

左点集为每组对战，源点流入的容量为a_{i,j}。

右点集为每个选手的胜场，流向汇点的容量为至多还能赢多少场。

看源点是否满流。

ASC38 J Jackpot

n个点m条边的无向图，A和B在其中博弈，A和B两人轮流行动，A为先手。

在第i轮行动中，那位玩家需要将i号点染成黑色或者白色。

如果某位玩家行动之后，图中存在一条边，两个端点被染成了同样的颜色，那么他就输了。

如果整张图全部染色后也没有以上的情况，则视为平局。

求博弈结果

$n < 1e5$, $m < 1e5$

ASC38 J Jackpot

考虑游戏的过程

在这个游戏中，染色的点将会形成一个个连通块。每染色一个点，都会带来一个新的连通块，并且将若干连通块连为一体。

对于每个连通块来说，它的染色情况由标号最小的点来决定，由此可以推断每个连通块分别是哪个玩家的势力范围。

ASC38 J Jackpot

考虑新染色一个点，它周围的染色情况

1、没有任何颜色，合法

2、与一块连通块相连，判断所连接的点颜色是否一致

3、与多块连通块相连，首先判断每块连通块是否合法其次，

ASC38 J Jackpot

其次,

a、如果都是自己的势力范围，没有问题

b、存在一个敌方势力，但它是最早的，没有问题

c、存在至少一个地方势力不是最早的，不合法 ,

因为对方可以针对你染色

可以用并查集维护

ASC37 A Achromatic

用k种颜色对长度为n的环进行染色，一种染色方案被称为achromatic是指：

将所有边两个端点的颜色对(c_i, c_j)进行统计，所有k*(k - 1)/2种可能的无序对都出现过。

现在对于给定的n，求最大的k，使得存在achromatic的染色方案，并且给出相应的构造。

$n < 1000$

ASC37 A Achromatic

奇思妙想

如果把颜色作为顶点，颜色对作为边的话，相当于在k个点的完全图上找一条长度为n的回路，使得每条边至少被经过一次。

寻找这样的最短回路的问题被称为中国旅行商问题，这是一个非常经典的问题，在一般图上需要用一般图最大权匹配算法，但是在完全图上，我们可以自己构造。

如果k为奇数，那么可以直接构造欧拉回路。
如果k为偶数，那么增加k/2条匹配边，使得每个点度数为偶数，构造欧拉回路。

ASC37 A Achromatic

构造方法

用一个个等差序列来遍历每个剩余系

一开始公差为1，如果遇到遍历过的颜色对，则增大公差，遍历下一个同余类。

例子

n = 5

1 2 3 4 5 1 3 5 2 4

ASC37 E WordCover

定义字符串的覆盖

x能够覆盖y，指的是x在y中可能出现位置的并集为全集。

例如aba可以覆盖abaababa

现给出长度为n的字符串，对于它的每个前缀，求其最小的覆盖串。

$n < 1e5$

ASC37 E WordCover

性质

1、所有的覆盖串，如果不是字符串本身，就是其border。

2、如果x能覆盖y，y能覆盖z，那么x也能覆盖z。

ASC37 E WordCover

设计DP

f[i]表示前缀i的最小覆盖串的长度，g[i]表示长度为i的border最远能连续覆盖到什么地方。

考虑一个新的前缀i

1、若g[f[fail[i]]] + f[fail[i]] >= i，则 f[i] = f[fail[i]]

2、否则f[i] = i

时间复杂度O(n)

ASC36 B Divisible Substrings

给定一个数字字符串s，询问有多少子串可以被n整除，字串是连续的。 $n < 30$

字符串很长，由m个上下文无关的产生式来生成，每个产生式左侧都为单独的
大写字符，右侧为大写字符、数字构成的字符串，并且其中的大写字符都大于左侧。

对于每个大写字符，只会在产生式的左边出现一次。每条产生式的长度不会
超过100。

$m < 26$

ASC36 B Divisible Substrings

例子

n = 2

s = 12312301231230

divisible.in	divisible.out
2 1000000000 3 A->BB B->CC0 C->123	46

ASC36 B Divisible Substrings

考虑简单版本的问题

如果直接给出数字串，则可以直接DP

f[i, j] 表示前i位以i为结尾的字符串中，余数为j的方案数

g[i]表示前i位中的总方案数

f[i, (j * 10 + s[i]) % n] += f[i – 1, j]

g[i] += f[i, 0]

ASC36 B Divisible Substrings

考虑产生式的解决，怎样求出最终的字符串是多少

产生式事实上形成了一个有向无环图，可以倒推求解。

对于每个字母，求解它的位数和取模之后的余数。

ASC36 B Divisible Substrings

考虑最后问题的解决

产生式至多由26条，但最终形成的字符串可能有100^26的规模，显然不能暴力。

这种进行常系数齐次线性转移的DP，应该考虑到使用矩阵乘法。

ASC36 B Divisible Substrings

考虑最后问题的解决

把我们要求的f,g合成一个列向量x，每次转移相当于乘上一个特定的矩阵A, y = A * x。

对于所有的数字，A都可以预处理得到。

对于产生式
X -> x_1 x_2 x_3 ... x_m

X的转移矩阵则为
A(x_m) * ... * A(x_2) * A(x_1)

ASC36 C Prepare Documents

公司现在需要准备n份文件，有两位秘书同时着手准备。

每份文件都需要1个小时，在任意时刻都可以两位秘书都可以切换手头的任务，而相应的准备的进度则不会丢失，但是两位秘书却不能同时准备同一份文件。

另外有m条约束，a_i 文件必须在 b_i 文件开始准备之前准备完毕，这些约束条件不会形成环。

至少花多少时间，可以准备完所有的文件。

$n < 1e5$, $m < 2e5$

ASC36 C Prepare Documents

例子

n = 6, m = 5

documents.in	documents.out
6 5 1 4 2 4 3 4 4 5 4 6	3.5

The following process allows to prepare all documents in 3.5 hours.

Time	Secretary 1	Secretary 2
0h00m	Start preparing document 1	Start preparing document 2
0h30m	Stop preparing document 1, switch to document 3	
1h00m		Finish document 2, start to continue preparing document 1
1h30m	Finish document 3, start preparing document 4	Finish document 1
2h30m	Finish document 4, start preparing document 5	Start preparing document 6
3h30m	Finish document 5	Finish document 6

ASC36 C Prepare Documents

考虑答案的性质

之所以会有实数，是因为为了协调各个文档之间的进度，让最晚结束的任务尽量早完成。

而我们只有两个人，因此更细的划分是没有意义的，将时间划分成半小时即可。

在拓扑图上将每个点拆点即可，一个为入度，一个为出度。

ASC36 C Prepare Documents

对有向无环图进行拓扑排序，每次同时选择两个点。

需要注意的是，我们优化的目标最晚的文件尽量早结束，因此我们文件的优先级定义为从他开始的最长链的长度。

用一个优先队列维护即可。

复杂度$O((n + m) \log n)$

ASC36 E Hot Potato Routing

HPR是一种网络传输信息的方式

该网络由n个节点，第i个节点有自己的传输协议$A_{i, j}$和协议长度L_i，
表示T时刻时该节点内如果有信息，则会花费一个单位的时间，传输到
$A_{i, T \bmod L_i}$节点中去。

现在有p条讯息可供选择，第i条讯息会在t_i时刻在网络中的s_i节点产生，
并且目标为d_i节点。如果它成功的到达了目标点，他就会立刻消失。
如果两条讯息在同一时刻处于同一节点，他们都会消失。

请选择发送尽量多的讯息，使它们都能到达目标点。

$n < 100$, $p < 100$, $L_i < 8$

ASC36 E Hot Potato Routing

如何判定相撞

由于T需要进行取模，所以事实上一条讯息只有n*2*2*3*5*7种状态，进行这么多步模拟之后，就能确定是否能到达终点。

两条讯息相撞的充要条件是
1、他们最后时刻所在的节点相同
2、产生的时间差和路程差相同

ASC36 E Hot Potato Routing

考虑贪心

1、显然我们只会选取能到目标点的信息，而这种信息的存活时间是有限的。

2、两条信息要相撞，他们的存活区间必须有交集。

按照消失的时间排序，贪心构造即可。

ASC36 H Restoring Permutation

给定一个长度为2n的全排列，拥有以下两种性质

1、不动点的个数恰好为n，$a_i = i$
2、每个偶数位置，都比它相邻的奇数位置的数要小

现在将所有的不动点都删去，将剩下的n个数重新离散化，成为一个新的n的全排列。

现在给出这个新的全排列，请给出一种原排列的合法解，或者判定无解。

$n < 1e5$

ASC36 H Restoring Permutation

例子

restore.in	restore.out
3 2 3 1	3 2 6 4 5 1
1 1	-1

ASC36 H Restoring Permutation

分析新排列的性质

新排列不可能有不动点，因为在原序列中，抛开不动点，剩下的位置形成了一个自闭的错位排列。

因此无解的充要条件是新排列中有不动点。

因此如果对新排列进行环分解，可以得到若干个非退化的环，也就是元素之间可以首位相连的匹配起来。

ASC36 H Restoring Permutation

考虑直接构造

将2*i, 2*i – 1两个位置匹配成一对，规定这两个位置中恰好有一个是不动点，则不动点的条件可以满足。

对于这两个位置，将从2*i, 2*i – 1两个数中选一个作为不动点，从2 * a[i], 2 * a[i] – 1中选一个作为错位点。

具体的选法要讨论a[i]和i的关系。

ASC36 H Restoring Permutation

若a[i] < i, 则令 2 * i - 1为不动点

若a[i] > i, 则令 2 * i为不动点

至于另外一个数，则看a[i]位置的不动点是怎么构造的，把剩下的那个拿过来。

ASC35 F Graph Factorization

一个图被称为是r-正则的，当且仅当它的每个点度数都为r 。

给出一个2n个点的完全图，要求将它的边分成m个集合，集合之间没有交集，且每个集合所形成的子图都是a_i-正则的。

保证$\sum_i a_i = 2 * n - 1$

$n < 100$, $m < 100$

ASC35 F Graph Factorization

例子，将K6分解

factor.in	factor.out
3 3 1 2 2	1 4 2 5 3 6 1 2 2 3 3 4 4 5 5 6 6 1 1 3 3 5 5 1 2 4 4 6 6 2

ASC35 F Graph Factorization

朴素的想法

先将完全图分成2 * n – 1个1-正则的图，然后将他们按照题目要求合并即可。

1正则的图就是一个2n个点的完备匹配。

ASC35 F Graph Factorization

第i个1-正则图为G_i, i < 2 * n

规定G_i的标识为其中存在1到i的边。

由于图中点的个数为偶数，可以将1和i去掉之后，将所有关于i对称的两点一一连边。

ASC38 C Coins

n个男孩分m个金币，分金币的方案如下

从1号男孩开始，1号先提出议案，所有人进行表决，如果全部通过，则按照该提案来分。如果至少有一人反对，则要从总金币中扣去d个金币。

之后则由2号进行提案，如此循环往复，直到没有金币，或者有一个提案得到通过。

每个男孩的策略很简单，假设当前提案他所获得的金币是x，如果他拒绝提案将不会再得到x，则他会赞成提案，否则反对。

求最后每个男孩得到多少金币。

$n < 1000$, $m < 1e18$, $d < 1e4$

ASC38 C Coins

样例

coins.in	coins.out
5 20 3	3 4 3 4 3
10 100 1	10 10 10 10 10 10 10 10 10 10
2 11 1	5 5

ASC38 C Coins

考虑最终状态

假设只剩$x <= d$个金币，现在由一号提案

若$x < n$, 则最后每个人都一无所获

若$x >= n$, 则1号可以给剩下每个人1个金币，剩下的自己独吞。

大家显然都会通过这个提案。

ASC38 C Coins

考虑DP

$f_{i,j}$表示现在剩下i个金币，从1号开始提案，则j最后会获得多少金币

显然$i < d$的情况都确定了。

考虑$i >= d$，那么将从$i - d$的情况递推而来

ASC38 C Coins

考虑i >= d，那么将从i – d的情况递推而来

如果相比于i – d的结局，它可以让每个人都多得1个金币，则它可以独吞剩下的所有钱。

否则的话，则f_{i, j} = f_{i – d, j -1}

从i = m % d开始模拟，因为只要考虑模d的同余类，所以事实上O(nd)也是可以的。

ASC35 H Robot Art

给定两个序列a_i, b_i, 每个位置都是1到k之间的整数，长度分别为n和m。

求最短的序列c，使得c既不是a的子序列，也不是b的子序列。

$n, m, k < 5000$

ASC35 H Robot Art

例子

robots.in	robots.out
2 5 1 2 1 2 1 5 2 1 2 1 2	4 1 1 1 1

ASC35 H Robot Art

考虑一个子序列的匹配过程

令f[i, j] 表示a_i之后第一个为j的位置，一个子序列的匹配就是从头开始不断后跳

如果只有一个a_i，而没有b_i?

贪心，每次都选取后面出现的最晚的那个数字。

ASC35 H Robot Art

二维的情况

令dp_{i, j}为长度为i的子序列，A序列匹配到j，B序列最远能匹配多少。

枚举下一位数字进行转移。

dp_{i + 1, f[j, k]} = max(g[dp_{i, j}, k])

分析复杂度

dp_{i, j}有意义在于要将前i个数分成j段，且每段最后一个数字在段中都是唯一的，故而状态有限。

ASC32 Minimal Matrix

给定一个n*n的字符循环矩阵，每个位置都是a或者b。

对其进行循环的上下左右平移，使得其字典序最小。

矩阵的字典序定义为将其按行展开后得到的字符串的字典序。

n <= 512

ASC32 Minimal Matrix

例子

matrix.in	matrix.out
baba baab abba bbba	aabb abbb abab bbaa

ASC32 Minimal Matrix

考虑一个循环矩阵的构成

每一行都是一个循环串，整个矩阵则是将这些小循环串看成一个个字符循环链接而成的大循环串。

考虑平移带来的效果

左右平移，则每个循环串对应的循环移动

上下平移，则每个小循环串不变，大的字符串整体循环移动

ASC32 Minimal Matrix

考虑问题的简单形式

如果只有一个循环串，就是求循环串的最小表示

ASC32 Minimal Matrix

如果枚举开始位置暴力，复杂度O(n^2)

考虑正在比较字符串从i和j开始的两个循环串

若S[i, i + k] = S[j, j + k]，且S[i + k + 1] > S[j + k + 1]，则显然[i, i + k + 1]
中不会有答案串。

于是每次尝试匹配比较，若失配，则将较劣的那段区间跳过，复杂度线性。

ASC32 Minimal Matrix

考虑二维的情况

先枚举左右平移的情况，这样每个循环串的形式就确定了。

在将每个小循环串看成一个整体，在大串中求循环串的最小表示。

总复杂度O(n^3)

ASC32E Points and Lines

n个点m条边的无向图G，Alice和Bob做游戏

两个人轮流行动，Alice先手，Bob后手，每次行动都可以将两个没有直接相连的点用一条边连接起来。如果行动之后整张图联通，则其获胜。

两个人都按照最优策略行动，求游戏结果。

$n < 150$, $m < C(n, 2)$

ASC32E Points and Lines

考虑终止状态

获胜的那个人肯定是将两个连通块连起来。

失败的那个人肯定是将三个连通块变成两个。

在此之前，两个人肯定将三个连通块内部的边全部连玩了。因此，关键的是剩下每个连通块内部可以连的边的数量。

ASC32E Points and Lines

考虑策略

对于剩下边数为偶数的连通块，最优策略中肯定不会连接它内部的边。

推论

对于剩下变数为奇数的连通块，最优策略中也至多连一次。

相应的，可以把每个连通块只考虑奇偶性，我们其实只关心总边数的奇偶性。

ASC32E Points and Lines

设dp_{i, j, k}为有i个奇数连通块，j个偶数连通块，剩下的边数为奇偶性为k

四种决策

两个奇数连边, dp_{i – 2, j + 1, k}

两个偶数连边, dp_{i, j – 1, k ^ 1}

奇偶连边，dp_{i, j - 1, k ^ 1}

对于k=1，dp_{i, j, 0}

计算sg函数即可

ASC34F Coins Game

n行m列硬币，Alice和Bob做游戏

两个人轮流行动，每次行动分两步，第一步，选择一个(x, y)处正面朝上的硬币，将其反转，在选择$0 <= a < x, 0 <= b < y$，继续将(a, b), (a, y), (b, x)三处的硬币反转。

当某位玩家无法行动时，该位玩家输掉游戏。

输入硬币的初始状态，求解最后的结果。

$n, m < 50$

ASC34F Coins Game

考虑只有一个正面朝上的硬币的情况

将其翻面后，会新产生三个正面朝上的硬币。

事实上，我们可以假装这三个硬币在三个不同的棋盘上，因为即使后续在相同的位置有重叠，重叠的两枚硬币等于没有，对胜负没有影响。

因此，这是三个独立的子游戏。

ASC34F Coins Game

SG函数，描述当前局面的整数函数。

对于独立的子游戏，和游戏的SG函数是子游戏SG函数的异或和。

对于某个局面，该局面的SG函数是所有后继局面SG的mex。

mex即最小的没有出现在集合里的整数。

ASC34F Coins Game

$sg_{x, y}$ 表示仅(x, y)位置上有正面朝上硬币的SG值

枚举$0 < a < x$, $0 < b < y$ 进行转移

将初始局面中的所有子游戏求异或和

ASC34J The Wall

平面上n个白点，m个黑点，要求画一个圆将黑点和白点分开。

$n,m < 120$

ASC34J The Wall

例子

wall.in	wall.out
2 2 0 0 0 1 1 0 1 1	YES 0 0.5 0.5
2 2 0 0 1 1 1 0 0 1	YES 0.5 0.5 0.7071067811865475 2
4 2 0 0 2 2 2 0 0 2 1 1 5 5	NO

ASC34J The Wall

朴素但错误的想法

至少有两个或者三个点在圆上，枚举圆上的点的组合，暴力。

事实上可能会使用更大的覆盖圆来调整位置。

ASC34J The Wall

假设圆内的是白点

考虑一对白点和黑点，所有合法的圆心位置，就是平面上到白点距离小于到黑点距离的点的集合。事实上，就是这两点中垂线的白点一侧，是一个完整的半平面。

枚举每一对黑白点，构造半平面的约束，半平面求交。如果交集非空，则选其中一点构造；否则无解。

复杂度O(nm (log n + log m))

ASC16J Yet Another Minimal Triangle

给定二维平面上n个点，找出三个点使得所形成的三角形面积最小

$n < 1000$

ASC16J Yet Another Minimal Triangle

正解比较复杂，这里我们介绍一个比较巧秒的做法

分析最优解的性质

1、最小的三角形内部不会再有其它点

2、三角形面积=1/2 * 水平宽 * 铅锤高

那个宽和高必有一个是较小的值

ASC16J Yet Another Minimal Triangle

猜想：存在一种刚性旋转变换，使得答案的三个顶点在水平序上接近

随机旋转角度，将点按照旋转后的水平坐标暴力，在相邻20个中暴力

多随机几次即可

复杂度 T * n * C(20, 2)，T为随机次数

ASC6F Crazy Painter

给定大小为n*m的矩阵，现在要对其进行涂色，涂成目标状态。

可以进行的操作:

1、花费h把一段水平连续的格子涂成一种颜色

2、花费v把一段垂直连续的格子涂成一种颜色

3、花费s涂一个单独的格子

限制条件是不能把一个格子先后涂成两种颜色

求最小花费 $n, m < 200$

ASC6F Crazy Painter

如果矩阵只有一种颜色的需求，剩下都随意，则是经典最小割问题

对于一个需要涂色的点(i, j)来说，它只有三种选择

1、被行i覆盖，费用为h

2、被列j覆盖，费用为v

3、被自己单独覆盖，费用为s

因此可以最小割建模，S到左点i容量为h，右点j到T的容量为v，
i到j的容量为s。

ASC6F Crazy Painter

对于多种颜色的情况，如何考虑?

1、因为一个格子不能被染成不同的颜色，因此各颜色的活动空间是独立的

2、同一行、同一列不能一笔画完了，因此对于每一段连续的空间都要单独建点。

时间复杂度$O(n^2m^2)$

ASC2F Road

n个点m条边的带权无向图，每条边边权为c_i，要求将边权改为d_i，使得前n-1条边形成的树是最小生成树。

同时$\sum |c_i – d_i|$最小。

$n < 200, m < 400$

ASC2F Road

考虑最终方案，显然我们不会增加前n-1条边的边权，也不会减小其它非树边的边权。

对于$i < n$，$d_i = c_i - l_i$

对于$i >= n$，$d_i = c_i + l_i$

$l_i >= 0$

ASC2F Road

由于前n-1条边是最小生成树，所以对于任意一条非树边j>=n，以及被j在树上横跨的树边i < n，都有d_i <= d_j

$c_i - l_i <= c_j + l_j$

$c_i - c_j <= l_i + l_j$

这是最小带权点覆盖的模型，给每个点分配点权满足覆盖约束，使得总点权最小。

ASC2F Road

但是特殊的是二分图

对于二分图，这可以用对偶问题求解，让$c_i - c_j$作为边权，l_i, l_j作为顶标，转化为二分图最大权匹配问题。

复杂度$O(n^3)$

ASC31E Yet Another Game With Words

Alice和Bob做游戏，从空字符串开始，轮流操作，每次可以在开头或者结尾添加一个小写字母。

如果某次操作之后，出现了禁止串，或者当前字符串不是任何一个禁止串的子串，则操作的玩家输掉游戏。

给定n个禁止串，每个串长度都不超过m，求解游戏胜负。

$n < 200, m < 50$

ASC31E Yet Another Game With Words

考虑中间状态

由于不能出现非子串，因此所有的中间结果必然是某个禁止串的子串之一。

考虑终止态

玩家不采用非子串的策略，并不会影响游戏的胜负，可以认为所有的终止态就是各个禁止串本身。

ASC31E Yet Another Game With Words

对于终态确定的博弈如何求解

反向BFS

将终态放入队列进行反向推测，如果一个状态被一个必败态访问到，则为必胜态。如果一个状态的先驱都是必胜态，则其本身为必败态。

ASC31E Yet Another Game With Words

在本题中，状态的转移是有向无环的，因此每个状态的结果都是确定的。

对于某些场景，并不是所有状态都能推断。

进行反向的BFS，对于那些没有BFS到的状态，则为未定态，即游戏达成平局，因为玩家们总能在未定态之间相互转移。

经典问题：猫鼠游戏

复杂度O(n m^2)

ASC31 I No Term Repitition Exact One SAT

给定一个CNF，求解是否存在一种赋值方式，使得每个子句中恰好有一项为真，其它都为假。

在CNF中，共有k个字句，n个变量，每个变量及其否定各自不会出现超过一次。

一个Conjunctive Normal Form形式如下

$$\Psi(x_1, \ldots x_n) = (t_{11} \vee t_{12} \vee \ldots \vee t_{1s_1}) \wedge \ldots \wedge (t_{k1} \vee t_{k2} \vee \ldots \vee t_{ks_k})$$

n < 200, k < 300

ASC31 I No Term Repitition Exact One SAT

一个例子

sat.in	sat.out
4 2 3 1 2 -3 3 -1 3 4	YES 1 0 1 0
5 4 3 1 2 3 3 -1 4 5 2 -2 -4 2 -3 -5	NO

ASC31 | No Term Repitition Exact One SAT

先考虑可行性，不考虑唯一性

因为每个变量及其否定最多出现一次，所以至多在一个子句贡献为真。

相应的，每个子句中为真的项对应的变量互相之间没有交集。

可以转化为匹配问题，每个子句只需找到一项令它为真即可。

ASC31 I No Term Repitition Exact One SAT

在考虑唯一性

每个子句中只能恰好有一项为真，其余都为假。

对于每个只出现一次的变量，单独新增一个子句，其中只有他没出现过的那种形式，转化为完备匹配的问题。

复杂度 O(nk)

ASC30 E New Mayor

给定n个点m条边的无向图，对其进行三染色，使得每个相连的点颜色都不同。

对于每个节点，它的所有邻居的导出子图都是连通的。

$n < 500$, $m < 10000$

ASC30 E New Mayor

mayors.in	mayors.out
5 8 1 2 1 3 1 4 1 5 2 3 3 4 4 5 2 5	Plan OK RGBGB
4 6 1 2 1 3 1 4 2 3 3 4 4 2	Plan failed

ASC30 E New Mayor

假设存在一组合法解

对于只有一个邻居的节点，它的颜色随意。

对于邻居大于等于两个的节点，它的邻居中至少有两种颜色，因此它的颜色是确定的。

ASC30 E New Mayor

从最开始的两个节点出发，逐一确定整张图的颜色。

如果一个节点的邻居颜色都被确定了，则它的颜色也可以得到。

总能找到一个节点它的邻居都被染上了颜色。

复杂度O(nm)

ASC30 E New Mayor

从最开始的两个节点出发，逐一确定整张图的颜色。

如果一个节点的邻居颜色都被确定了，则它的颜色也可以得到。

总能找到一个节点它的邻居都被染上了颜色。

复杂度O(nm)

ASC30 I Segment Transformation

给定两个长度相等的字符串s,t，由ACGT四个字母组成，每次可以选取一段子串，将其中的字符循环变换一次。

求解至少变换多少次，才能将s变为t，并输出方案。

字符串长度不超过100。

ASC30 I Segment Transformation

不考虑循环变换，则为经典问题。

对于数列a_i，每次可以选择一个区间减去某个数，问最少需要操作多少次，可以将整个数列变为零。

求解方法为并查集，自大到小合并区间。

对于[x, y, p]和[y + 1, z, q]两个操作，考虑p和q的关系。

ASC30 | Segment Transformation

若p=q，则直接合并操作。

若p<q，则将操作改为[x, z, p]和[y + 1, z, q - p]。

反之则对称构造。

可以用并查集维护连续区间的合并。

ASC30 I Segment Transformation

对于循环的情况，可以假装没有循环。

因为在模4的剩余系中操作，如果每个位置上循环的距离相差达到了4，
则可以让大的一部分全部减去4，依然是合法的解。

复杂度O(n log n)

ASC10 C Order Reserving Coding

给定n个单词的出现次数，求一种编码方式，满足以下条件

1、是前缀编码

2、编码长度最短

3、序号小的单次，编码字典序小于序号大的单词。

$n < 2000$

ASC10 C Order Reserving Coding

只考虑前两条，则为经典的哈夫曼编码。

考虑哈夫曼树生成的过程，每次选取两个子树，一个添加前缀0，一个添加前缀1。

现在新增两个性质，一是0/1的添加方式要和原来的序号大小一致，二是每个子树都是连续的序号区间。

即序号的顺序即为哈夫曼树的dfs序。

ASC10 C Order Reserving Coding

可以使用区间DP，f_{i, j}表示解决i到j的编码所需的最小花费。

f_{i, j} = min(f_{i, k} + f_{k + 1, j}) + sum_{i, j}

sum_{i, j} 表示i到j之间单词出现的总次数。

显然可以用四边形不等式进行优化。

O(n^3)

ASC10 H Periodic Tiling

给定网格图中的一个连通块，判定该连通块能否密铺平面。

网格图的大小为$n * m$, $n < 50$, $m < 50$

ASC10 H Periodic Tiling

例子

tilings.in	tilings.out
.... .*.. .**.	YES
.... .*.. .**.* ..***	NO
.... .***. ..*.. .***. ..*..	YES

ASC10 H Periodic Tiling

Tiling Theorem

所有的网格多边形平面密铺，事实上只有两种本质不同的模式：

1、和正方形密铺等价

2、和正六边形密铺等价

ASC10 H Periodic Tiling

和正方形等价

边界上顺序存在四个点ABCD，使得AB与CD全等，BC与DA全等。

和正六边形等价

边界上顺序存在六个点ABCDEF，使得AB与DE，BC与EF，CD与FA对应全等。

全等指的是中心对称的全等。

ASC10 H Periodic Tiling

换句话说，可以将周长分成长度相等的两半L_1, L_2。

再将内部分成2段或者三段，使得L_1和L_2中每段对应全等。

假设周长的一般为k，如果[i , j]和[i + k, j + k] 对应全等，则从i到j连接一条长度为1的边。

如果存在dist(i, i + k) <= 3, 则表明可以密铺。