2016 年集训队作业 Codechef 试题泛做

杭州学军中学 金策

(比较有意思的题目会标上 ★)

1 数据结构题

1.1

试题编号	AUG15 DISTNUM
试题名称	Simple Queries

题目大意	算法讨论
维护一个正整数数组。现在有关于它的 Q 个询问。询问有 5 种： （1）对于下标在 $[l,r]$ 区间的所有元素去重后构成的集合，求出 $\sum_{x<y<z} xyz$ （2）$a[x]=y$ （3）删除 $a[x]$ （4）在 $a[z]$ 之后插入 y 元素 （5）询问下标在 $[l,r]$ 区间内的有多少种不同的元素。 $(n,q \le 10^5)$ 可以离线。	首先有插入操作比较麻烦，但因为是离线的，可以事先预处理出每个插入的位置。这一步可以用一个平衡树来完成。这样后面只要在静态序列上搞就可以了。 操作 (5) 是经典的数颜色问题，只要把每个点看成二维平面上的点 $(lastpos[x],x)$，这样只要统计区间 $[l,r]$ 内 $lastpos<l$ 的点的数量就可以了。这可以用二维线段树来完成。 操作 (1) 中要求的那个式子可以通过维护一次方和、二次方和、三次方和来得到，然后也用上面的方法搞就可以了。

时空复杂度	空间 $O(n\log^2 n)$，时间 $O(n\log^2 n)$。

1.2

试题编号	MAY15 CBAL
试题名称	Chef and Balanced Strings

题目大意	算法讨论
一个小写字母组成的字符串。一个子串是平衡的当且仅当每种字符都出现偶数次。每次询问一个区间内所有平衡子串的长度的 $type$ 次方和。 强制在线，$(n \le 100000, type \in \{0,1,2\})$	给每种字母分配一个二进制位。求出数组的前缀异或和，那么子串 $[l,r]$ 平衡当且仅当 $pre[l-1]=pre[r]$。由于只涉及到相等关系，可以把这些数值离散化。 然后用经典的在线莫队方法：分块解决，预处理出第 i 块到第 j 块的答案。询问的时候先找出对应的大块区间，然后把两边多出来的部分往里面暴力添加。添加时只需要维护每种下标的 0 次方、1 次方、2 次方和即可，这些值也需要在开头预处理好，于是每添加一个数是 $O(1)$ 的。

时空复杂度	空间 $O(n\sqrt{n})$，时间 $O(n\sqrt{n})$。

1.3

试题编号	OCT14 TRIPS ★
试题名称	Children Trips

题目大意	算法讨论
一棵 n 个点的树，每条边长度为 1 或 2。每次询问 u,v,d，问 u 到 v 的路径最少要分成几段，使得每段的长度都不超过 d（不能在一条边中间断开）。 $(n \le 10^5, q \le 10^5)$	取一个 $S=O(\sqrt{n})$。任取一个根，然后将所有高度为 S 的倍数的点标为关键点，这样任意一条简单路径上只有 $O(\sqrt{n})$ 个关键点。 对于 $d>\sqrt{n}$ 的情况，至多分成 $\sqrt{n}$ 段，因此可以贪心暴力。 对于 $d \le \sqrt{n}$ 的情况，我们只要让它每次跳过一个前面的关键点，这样也只要跳 $O(\sqrt{n})$ 次。 为了实现跳跃，需要对于所有 $d \le \sqrt{n}$ 处理出每一个 u 跳到祖先的最近关键点所需要的步数，以及跳过后落在哪个位置。 如果按照原题的描述，看起来还要实现从 LCA 往下面跑的过程，其实不必要，这个贪心两边开始往中间分别贪也是没关系的，所以只要分别从 u,v 往上朝着 LCA 跑就可以了。 要实现暴力跳 $d>\sqrt{n}$ 步，可以转化为跳若干个大步，再用预处理的数组的结果跳一个小步。跳大步的总数是 $O(\sqrt{n})$ 的，所以复杂度还是 $O(\sqrt{n})$。

时空复杂度	空间 $O(n\sqrt{n})$，时间 $O(n\sqrt{n})$。

1.4

试题编号	FEB13 QUERY
试题名称	Observing the Tree

题目大意	算法讨论
维护 n 个点的无根树上的点权，支持：链加等差数列，询问链和，回到历史版本。 强制在线。 $(1 \le n, m \le 10^5)$	首先是树链剖分转化为链上的问题。 对线段树上的每个结点 $[l_i, r_i]$，除了维护 sum_i 之外，还维护标记 a_i, d_i。表示对这个区间已经加过了一个首项为 a_i，公差为 d_i 的等差数列。 再用熟知的可持久化线段树方法即可。为了减少常数，可以使用标记永久化。

时空复杂度	时间，空间都是 $O(n\log^2 n)$。

1.5

试题编号	MARCH14 GERALD07 ★
试题名称	Chef and Graph Queries

题目大意	算法讨论
n 个点的无向图。每次询问给出 l,r, 问仅保留编号在 $[l,r]$ 的边，图中有几个连通块？ $(n,m,q \le 200000)$	按照边编号顺序插入边，并用 LCT 维护一个生成森林。每次插入边 i 形成环时，删除环上编号最小的边 j，并令 $f[i]=j$；不成环则 $f[i]=0$；是自环则 $f[i]=\infty$。 假如现在刚插了边 i，就回答所有形如 $[l,i]$ 的询问，区间中 $f[x] \ge l$ 的边 x 都是废边，剩下的边每条都使连通块数减 1。这可以用线段树查询。

时空复杂度	空间 $O(n+m)$，时间 $O((n+m)\log(n+m))$。在线做法需要可持久化线段树，空间多一个 log。

1.6

试题编号	MARCH15 TREECNT2
试题名称	Counting on a Tree

题目大意	算法讨论
一棵 n 个点，带整数边权 a_i 的树，q 个操作，每次修改一条边的权，并查询有多少条简单路径上的权值 gcd 为 1。 $(n \leq 100000, a_i \leq 1000000, q \leq 100)$	容易想到容斥，那么需要枚举所有 $\mu(x) \neq 0$ 的 x，并选出边权为 x 的倍数的边组成的子图统计其路径数量然后加到答案里去。 如果每次都重复求一遍会很慢。注意到只有 100 个修改操作，也就是说大部分边都是保持不变的，所以在用并查集做的时候，可以把那些不会被删除的边给加进去，然后对于每个操作，暴力把需要插入的边插到并查集，然后统计答案，之后再恢复到插之前的状态，然后做下一个询问。这样复杂度是可以承受的。

时空复杂度	空间 $O(nd(a_i))$，时间 $O((nd(a_i) + (qd(a_i))^2)\log n)$。

1.7

试题编号	APRIL15 BWGAME ★
试题名称	Black-white Board Game

题目大意	算法讨论
考虑所有 n 排列 p，满足 $l_i \leq p[i] \leq r_i$，这些排列中奇排列和偶排列哪个数量更多？$n \leq 100000$	考虑一个 $n \times n$ 矩阵，第 i 行的 $[l_i, r_i]$ 列填 1 ，其他填 0，问题就转化为求这个矩阵的行列式。 考虑消元的过程，当前处理第 j 列，则把所有左端点为 j 的区间拿出来，取其中右端点位置最靠左的 p，并用 p 去消其他左端点为 j 的区间，容易发现消元之后每一行的 1 仍然保持连续一段区间的形状，而且被消的区间的左端点变为 p 的右端点 +1，而其各自的右端点不变。于是可以用可并堆来维护这些询问与操作。

时空复杂度	空间 $O(n)$，时间 $O(n \log n)$。

1.8

试题编号	JAN15 XRQRS
试题名称	Xor Queries

题目大意	算法讨论
维护一个数列，支持以下操作： （1）在末尾插一个数； （2）求区间内的数与 x 异或的最大值； （3）删除末尾的 k 个数； （4）求区间内一个数的排名； （5）求区间内第 k 大。 (操作数目 $m \leq 500000$，权值 $x \leq 500000$)	用可持久化 01trie 来维护。 插一个数就直接在 last 版本上插，删末尾几个只要将 last 指针往前移就行。然后 01trie 维护出现次数时是可以相减的，用 r 的减去 $l-1$ 的就得到 $[l, r]$ 的。所以（2）（4 ）（5）几个经典操作也是能做的。

时空复杂度	空间 $O(m \log x)$，时间 $O(m \log x)$。

1.9

试题编号	JUNE14 SEAARC
试题名称	Sereja and Arcs

题目大意	算法讨论
长为 n 的序列 a_i，有多少组 $i < j < k < l$ 满足 $a_i = a_k \neq a_j = a_l$。 $(n \leq 100000, a_i \leq 100000)$	我们按出现次数多的（大于 S 次）颜色和出现少的 ($\leq S$ 次）分类计算。 首先算 a_i, a_j 都是少数颜色的：注意到两个颜色各选两个有 $AABB, ABAB, ABBA$ 三种情况，$AABB$ 是很好统计的，所以只要算出 $ABBA$ 的情况即可，这个只要从左到右扫，扫到 A 的时候就枚举之前出现过的 A，于是只要查询区间内有几对 BB 就好，这可以用树状数组，每次扫到 B 的时候往前每一位 B 处增加 1 就行。这样要做不超过 $S^2 \times n/S$ 次树状数组操作。 接着算 a_i 是多数色的。总共只有不超过 n/S 种多数色，枚举固定一个多数色 A 然后算。算出每个 B 左边有多少个 A，两两乘起来，这样算得的结果包括了 $ABAB$ 和 $AABB$ 和 ABB，把后两类剔除即可。这样一次是 $O(n)$ 的。

时空复杂度	空间 $O(n)$，时间 $O(n\sqrt{n \log n})$ 。

1.10

试题编号	MAY14 ANUDTQ
试题名称	Dynamic Trees and Queries

题目大意	算法讨论
维护一棵树的点权，支持以下操作： （1）添叶子； （2）子树加； （3）删除一棵子树； （4）询问子树和。 强制在线。$(n, m \leq 100000)$	如果只有 (2)(4) 操作，就是一个经典的 DFS 序题目，用线段树维护即可。 现在问题在于树形态会变化。我们可以用一棵 splay 来维护树的 DFS 序。为了方便，采用括号序列。添叶子时，只要在父亲结点对应的左括号后面加一对括号。子树操作只要把一对括号之间的区间提取出来，然后进行操作。

时空复杂度	空间 $O(n)$，时间 $O(n \log n)$。

1.11

试题编号	NOV13 MONOPLOY
试题名称	Gangsters of Treeland

题目大意	算法讨论
一棵有根树，每个结点有颜色，初始时都不同。一条边若两端颜色不同则需要 1 的代价。支持两种操作： （1）将 u 到根路径上的点染成一种没出现过的颜色； （2）询问一个子树内每个点到根的代价的平均值。 $(n, q \leq 100000)$	根据题意，修改操作实质就是 LCT 的 access 操作，所以只需要用 LCT 维护这棵树，并在每次 access 时发生虚实边替换的时候，对应的修改边上的边权。 维护边权和询问可以转化到 DFS 序上，然后用树状数组/线段树来查询。

时空复杂度	空间 $O(n)$，时间 $O(n \log^2 n)$。

1.12

试题编号	SEPT14 QRECT
试题名称	Rectangle Query

题目大意	算法讨论
维护平面上的矩形（边平行坐标轴），支持： （1）插入矩形； （2）删除矩形； （3）询问与给定矩形相交的矩形有多少个。 $(q \le 100000)$	可以转化一下，统计与给定矩形不相交的有几个。这只要判断端点的关系就可以了。先看看一维情况，就是统计右端点右边的左端点数和左端点左边的右端点数。二维情况稍微推广一下，就是右下端点的左下角的左上端点数目（以及其他三个方向，通过旋转 90 度即可得到）。 这样题目就变成了加点、删点及二维区域点查询。经典做法是用二维线段树，或者离线分治套一维树状数组。

时空复杂度	空间 $O(n)$，时间 $O(n\log^2 n)$。

1.13

试题编号	MAY13 QTREE
试题名称	Queries on tree again!

题目大意	算法讨论
一个 n 个点 n 条无向边的连通图，边带有边权。支持两种操作： （1）将 u 到 v 路径上的边权乘 -1； （2）询问 u 到 v 路径上的边权的最大连续子段和。 路径都指最小结点数目的路径，保证唯一。 $(n \le 10^5, q \le 10^5)$	维护最大连续子段和的经典做法是记录四个信息 $sum, lmax, imax, rmax$，这里需要变号所以还要维护三个 min。这样就支持变号已经信息合并了。链询问和操作只要树链剖分并用线段树维护即可。 这里是一个基环外向树。比较方便的处理方法是看成整棵树加上一条多余边。对于给出的两点，只要算一下经过这条边的路径和不经过的哪个更短即可，如果走额外边更短的话，所需要的路径就拆成了两条树上路径加这条额外边。

时空复杂度	空间 $O(n)$，时间 $O(n\log^2 n)$。

1.14

试题编号	MARCH14 STREETTA ★
试题名称	The Street

题目大意	算法讨论
两个长为 n 的数组 $a[], b[]$。支持三种操作： （1）给 a 的区间 $[l, r]$ 加上等差数列； （2）给 b 的区间 $[l, r]$max 上等差数列； （3）询问单点的值。 $(n \le 10^9, q \le 3 \times 10^5)$	首先离散化然后用线段树维护。 第一个操作有经典的做法，对线段树每个结点标记两个值 k, b，表示这一个区间中的所有 x 都被加上了 $kx + b$。（永久化标记） 第二个操作也有经典的做法，对线段树每个区间维护一条（坐标系中的）线段。每当一条新的线段覆盖到这个区间上，先检查这条线段是否永远比原来的优（或劣），如果是则替换（或扔掉），如果不是则有交点，那么在交点的另一侧的儿子区间肯定是完全覆盖（或扔掉的），所以只要往交点这一侧递归下去就行。

时空复杂度	空间 $O(m)$，时间 $O(m\log^2 m)$。

1.15

试题编号	NOV14 FNCS
试题名称	Chef and Churu

题目大意	算法讨论
一个长为 n 的数列 $a[]$ 和 n 个函数。第 i 个函数返回值为数列中第 l_i 到 r_i 项的数字之和。 支持两种操作： (1) 将 $a[x]$ 修改为 y; (2) 询问第 p 个到第 q 个函数的返回值之和。 ($n \le 10^5$)	首先对操作分块。开始处理某一块操作前，把目前的数组情况以及每个函数的返回值（及他们的前缀和）先预处理好。接下来只需要考虑块内的修改操作对块内且在它后面的询问操作的贡献就行了。我们枚举块内的每一对修改操作和询问操作统计贡献，这样就变成了查询对于某个点 x ，有多少个 $p \le y \le q$ 满足 $l_y \le x \le r_y$，这可以变成两次询问 $1 \le y \le p-1$ 与 $1 \le y \le q$ 。 这样对于所有块共有 $O(n\sqrt{n})$ 个这样的询问。扫一遍 n 个函数，并用区间加法 $O(\sqrt{n})$，询问单点 $O(1)$ 的分块维护即可。

时空复杂度	空间和时间都是 $O(n\sqrt{n})$

1.16

试题编号	SEPT14 FIBTREE
试题名称	Fibonacci Numbers on Tree

题目大意	算法讨论
维护一棵树上的点权。支持操作： （1）链加 fib 数列； （2）询问链和； （3）定一个点为根后询问某点的子树和； （4）回到过去某一版本。 模 10^9+9 输出答案，强制在线。 $(0 \le n, q \le 100000)$	树剖后上线段树，线段树要可持久化。另外由于有子树询问，还需要 DFS 序，这个 DFS 序要按照先走重链的顺序，这样既可以处理链又可以处理子树。换根不需要真的换根，只要检查一下树根和询问点的祖先关系，看看询问的子树在原树中也是子树，还是子树的补集。 现在只需要考虑怎么用线段树维护 fib 数。按照 fib 数的通项公式，可以将它拆成两个等比数列的和，而等比数列是可以维护的，只要标记一个区间被加的等比数列的首项就可以了。 计算的时候涉及到 $\sqrt{5}$ 的，它是 10^9+9 的二次剩余，所以直接用整数就可以完成计算。

时空复杂度	空间 $O(n\log^2 n)$，时间 $O(n\log^2 n)$。

1.17

试题编号	JULY12 DGCD
试题名称	Dynamic GCD

题目大意	算法讨论
维护树上的点权，支持操作（1）链上加 d，（2）询问链上点权的 gcd。 $(n, q \le 50000)$	先树链剖分转化为链上问题。考虑链上的这个数列 $a_1, a_2, \cdots, a_k$，对它做差分得到 $a_1, a_2-a_1, a_3-a_2, \cdots$。这样修改操作就转化成了单点修改。 另外注意到 $\gcd(a_i, a_{i+1}, \cdots, a_j) = \gcd(a_i, a_{i+1}-a_i, \cdots, a_j-a_{j-1})$，于是只要查询区间的 gcd 以及 a_i 的值即可。a_i 就是一个前缀和。所以只要用线段树维护 gcd 和 sum 即可。

时空复杂度	空间 $O(n)$，时间 $O(n\log^2 n\log d)$。

1.18

试题编号	FEB14 COT5 ★
试题名称	Count on a Treap

题目大意	算法讨论
维护一个 treap。支持： （1）插入一个关键字为 key，权重为 w 的结点； （2）删除一个关键字为 key 的结点； （3）询问结点 u, v 在 treap 上的距离。 根结点权重最大。 $(n \le 2 \times 10^5)$	我们按照 key 作为下标的顺序，维护一个 w 的序列。给定两个位置 l, r，区间 $[l, r]$ 中权值最大的即为 l, r 的 LCA。要询问距离的话，只要知道这三个结点的高度即可。 注意到 $h(u) = u$的祖先数目 $= u$的左边的祖先数目 $+ u$的右边的祖先数目。于是可以左右分开统计。 x 是 y 的祖先当且仅当 x 到 y 这段路上没有比 x 更大的点把它挡住，所以这是一个经典的“看到几座楼房”问题。 经典做法是用线段树维护，除了维护区间最大值之外，还要记录从 l 中最高的结点往兄弟结点 r 方向能看到几个，以及从 r 中最高结点往 l 能看到几个。查询这个东西的复杂度是 $O(\log n)$，因为利用记录的信息可以每次只往一边递归。所以单点更新要做 $O(\log n)$ 次，复杂度是 $O(\log^2 n)$。回答一次高度询问需要对拆分成的 $O(\log n)$ 个区间分别查询，所以总复杂度也是 $O(\log^2 n)$。

时空复杂度	空间 $O(n)$，时间 $O(n \log^2 n)$。

1.19

试题编号	OCT14 BTREE ★
试题名称	Union on Tree

题目大意	算法讨论
一棵边权均为 1 的树，每次询问给定 k 个点 u_i 以及各自的控制距离 d_i，每个点能控制距离不超过 d_i 范围内的点，问被控制的点总共有几个。 $(n \le 50000, \sum k \le 500000)$	首先对询问点建出虚树，虚树上没有控制能力的点的 d 可以设为 -1。 考虑两个点 u, v，我们可以把 d_v 设置为 $\max(d_v, d_u - dis(u, v))$。我们对所有点按照这一规则更新它们的控制能力直到不能更新为止，这一步可以用一个类似 dijkstra 的过程来完成。 对于虚树上每条边 (u, v)，容易看出 u, v 的控制范围的交集是以 r 点为中心，$(d_u + d_v - dis(u, v))/2$ 为半径的区域，其中 r 是路径 (u, v) 上的某点（可以事先在每条边上加虚点保证中点存在）。然后答案即为每个点的控制点数减去每条边上 (u, v) 的控制范围交集的点数。（由于 d_u 都已经被更新到最大，所以这可以证明是正确的）。 为了统计离 u 距离不超过 d 的数目，可以用点分治。点分治的每层需要记录各个高度的点的数量的前缀和。

时空复杂度	空间 $O(n \log n + k)$，时间 $O(\sum k \log k + n \log n)$。

1.20

试题编号	DEC13 QTREE6
试题名称	Query on a tree VI

题目大意	算法讨论
维护一棵树，每个点有黑白两色之一，支持以下操作： （1）询问一个点所在的同色连通块的大小； （2）翻转一个点的颜色。 $(n, m \leq 100000)$	对于每个点维护以它为根结点的黑色连通块和白色连通块的大小 $b[i], w[i]$。 询问时，只要找出询问点所在的连通块的根 i，再输出 $b[i]$ 或 $w[i]$ 就可以了。找出根可以用树状数组 + 二分在 $O(\log^2 n)$ 时间完成。 翻转一个点的颜色时，从它到它所在连通块的根这段路径上的结点 v 的 $b[v], w[v]$ 值会发生变化，所以我们还需要实现一个链加操作。由于询问都是单点的，所以这里可以变成单点改，子树和查询，更加方便。

时空复杂度	空间 $O(n \log n)$，时间 $O(n \log^2 n)$。

1.21

试题编号	FEB12 FINDSEQ
试题名称	Find a Subsequence

题目大意	算法讨论
给一个整数列 $a[1..n]$，和一个 5 排列 $p[1..5]$。求出一个 a 的长为 5 的子序列，使得它的大小顺序和 p 一致。 $(n \leq 1000)$	枚举 $p[2], p[4]$ 分别对应哪两个位置，那么 $p[1], p[3], p[5]$ 各自属于一段区间，这三个区间之间不相交，处理起来更方便。 现在考虑这三个数怎么选，最大的那个肯定是越大越好，最小的那个也是越小越好。所以只要求出最大的和最小的，然后要找的是卡在中间的那个。这可以通过二分查找完成。为了二分的方便，可以预处理一个 $f[i][j]$ 数组，表示前 i 个数中不超过 j 的有几个。 预处理之前需要先离散化。

时空复杂度	空间 $O(n^2)$，时间 $O(n^2 \log n)$。

1.22

试题编号	JAN12 CARDSHUF
试题名称	Card Shuffle

题目大意	算法讨论
维护一堆初始顺序是 $1 \sim n$ 的扑克牌。每次执行下述操作： 从牌堆顶端拿走 A 张牌。再从牌堆顶端拿走 B 张牌。将第一步拿走的 A 张牌放回到剩下的牌堆上面。从牌堆顶拿走 C 张牌。将第二步你拿起的 B 张牌一张一张放到牌堆顶。最后，将剩下的 C 张牌放回到牌堆顶。 最后输出操作完后的序列。 $(n, m \leq 10^5)$	用数据结构模拟这些操作。需要支持的操作有区间裁剪并插入，以及区间翻转。用 splay 来实现即可。

时空复杂度	空间 $O(n)$，时间 $O(n \log n)$。

2 图，匹配，网络流

2.1

试题编号	MAY15 GRAPHCNT
试题名称	Counting on a directed graph

题目大意	算法讨论
n 个点的有向图。有多少对结点 (x, y) 满足存在 1 到 x 和 1 到 y 的路径，且两条路径仅有公共点 1。 $(n \leq 100000, m \leq 500000)$	以 1 为根求出 dominator tree。这样，树上 LCA 为 1 的每一对结点都符合条件。

时空复杂度	空间 $O(n+m)$，时间 $O((n+m)\log(n+m))$。

2.2

试题编号	FEB14 DAGCH
试题名称	Graph Challenge

题目大意	算法讨论
n 个点的有向图按 DFS 序编号。x 是 y 的 supreme vertex，当且仅当存在有向路 $x = v_0, v_1, \cdots, v_k = y$ ，且 $x \leq y \leq v_i$。 v 是 w 的 superior vertex，当 v 是 w 的所有 supreme vertex 中编号最小的。 输出每个结点有多少个结点将它作为 superior vertex。 $(n \leq 100000, m \leq 200000)$	按照定义，superior vertex 即为半必经点 semi。用 dominator tree 的算法计算即可。

时空复杂度	空间 $O(n+m)$，时间 $O((n+m)\log(n+m))$。

2.3

试题编号	MAY12 TICKETS
试题名称	Selling Tickets

题目大意	算法讨论
一个无自环，可能有重边的无向图。求一个 k，使得不论怎样选取其中 k 条边，都能让这些边各自选择自己的一个端点，且每条边选到的点不重复。 $(n \leq 200, m \leq 500)$	根据题意，我们要选择一个边数最少的子图，使得这个子图的点数 = 边数 −1。 这样的图一共有三种情况： （1）两个点之间的三条路径。这可以枚举起点和终点，然后跑 BFS，并记录下到达终点的三条距离。 （2）一条路径连着两个简单环。这只要对于每个点求出经过它的最小简单环就可以了。这可以 BFS，如果一条非树边的两端的 LCA 是根结点，则用它来更新答案。顺便要记录一下环上是那些边。因为最后合并答案时不能有两个相同的环并起来。 （3）一个点连着两个简单环。只要顺便记下通过每个点的次短环就好了。 中间可能会有一些重复边经过，但经过重复边的一定不是最优解，只要无视它们就行。

时空复杂度	空间 $O(n+m)$，时间 $O(n^2(n+m))$。

2.4

试题编号	JAN14 TAPAIR ★
试题名称	Counting The Important Pairs

题目大意	算法讨论
给一个简单连通无向图，有多少对边，使得删除这两条边后图会不连通？ $(n \leq 100000, m \leq 300000)$	这题有一个经典的 hash 做法：求一棵 dfs 生成树，并给所有非树边（只能是返祖边）分配一个随机权值；所有树边的权值是覆盖了它的非树边的权值的异或和。一个边集被删去后使得原图不连通，则它存在一个边的子集使得其权值的异或和为 0。 用这个做法，此题中只需考虑 2 元边集，所以更加方便。

时空复杂度	空间 $O(n+m)$，时间 $O(n+m\log m)$。

2.5

试题编号	JUNE11 MINESREV
试题名称	Minesweeper Reversed

题目大意	算法讨论
一个 $r \times c$ 的扫雷盘面。开始时所有格子都是开的，你要把它们关上，且用的步骤最少。可以点击一次关闭一个格子，同时关闭所有可以和它同时打开（按照正常规则）的格子。$(r, c \leq 50)$	这是一道阅读理解题。 首先雷是要一个个点的。然后考虑周围有雷的数字格和周围无雷的空白格。空白格的连通块可以一次性打开，还顺便打开了周围的数字格。一个数字格旁边至多有两个不同的空白格连通块，如果有两个，就可以通过打开这个数字格然后把他俩都打开。所以可以在它们之间连一条边。 接下来求出这个图的最大匹配数。对于所有匹配边，一次性打开这两个块。剩下的块都一个一个的开。 由于这里的图不一定是二分图，所以求最大匹配要用带花树算法。

时空复杂度	空间 $O(rc)$，时间 $O((rc)^3)$。

2.6

试题编号	JULY15 HAMILG ★
试题名称	A game on a graph

题目大意	算法讨论
无向图 G 上有一颗棋子，初始时放在结点 v 上。两个人轮流移动棋子，可以将它沿着边移到相邻结点上，但不能移到它曾经走到过的结点。不能走的人输。 统计有多少个初始结点 v 能使得先手胜利。 $(n \leq 2000, \sum m \leq 10^6)$	首先注意到一个结论：v 能使得先手胜当且仅当存在某个 G 的最大匹配不包含 v。（考虑不存在增广路这一性质即可证明） 先用带花树算法跑一遍最大匹配。然后再从所有没被匹配的结点开始做一遍寻找增广路的过程，这个过程是找不到增广路的，但是在途中入队过的结点都可以不被最大匹配包含。输出入队过的结点数目即可。

时空复杂度	带花树的时间复杂度 $O(nm)$。

2.7

试题编号	AUG12 GTHRONES
试题名称	A Game of Thrones

题目大意	算法讨论
一堆数字写在纸上。两人博弈。第一个人随便擦一个数字。后一个人也要擦一个数字，但擦的数字必须和对方上次擦的数字只相差一个素因子。不能动的人就输。 求必胜方。如果先胜的话要输出第一步操作的可行步骤。 $(1 \le n \le 500, 1 \le number \le 10^{18}, 1 \le count \le 10^9)$	这题是 HAMILG 那题 (undirected vertex geography) 的特殊化。 这里的图显然是二分图。但是由于一个数可以出现很多次，建多个点是无法承受的，所以要把匹配写成最大流的形式。 删一个点后判断最大匹配数是否减少，如果重复跑最大流会太慢，比较好的办法是利用退流。退流 (u,v) 时只要让 (u,v) 的流量减 1，然后从 t 到 v 增广，从 u 到 s 增广。 构图时需要质数判定，可以用 miller-rabin。

时空复杂度	空间 $O(n)$，时间 $O(n^2 \log A + maxflow(n, n^2))$。

2.8

试题编号	JULY14 GNUM
试题名称	Game of Numbers

题目大意	算法讨论
整数 a 序列和 b 序列，每次挑出 i,j,p,q，其中 (i,j),(p,q) 分别不出现过，且 $a[i] < b[j], b[p] < a[q], \gcd(a[i], b[j], b[p], a[q])) > 1$。最多能取几次。 $(1 \le n \le 400, a[i], b[i] \le 10^9)$	这是一个二分图最大匹配模型，对所有 $a[i] < b[j]$ 建一排点，$a[i] > b[j]$ 的 (i,j) 建另一排点，若 gcd 不为 1 则连边。 但这样构图边数太多，需要优化。首先将 $\gcd(i,j) = 1$ 的点扔掉，再将 gcd 相同的点合并。然后优化边，两个数有边当且仅当有公共质因子，那么就对所有质因子建一个点，然后把拥有这个质因子的数字连向它（或者由它连出）。然后跑 dinic 就可以了。

时空复杂度	网络流点数 $O(n^2)$，边数 $O(n^2 \log a_i)$。

2.9

试题编号	DEC14 RIN ★
试题名称	Course Selection

题目大意	算法讨论
有 n 个科目和 m 个学期。第 j 个学期上 i 科目的得分是 $x[i][j]$。有 k 个条件 (a,b) 表示 a 科目需要在 b 科目之前上。安排每门课在哪个学期上，获得的最大总分是多少？$(n, m, k \le 100)$	最小割建模。 对每门课建一排共 $m+1$ 个点，在第 i 个和第 $i+1$ 个点之间割断，表示在第 i 个学期上这门课。割边的权值设为这门课的最大可能得分减去这个学期的得分。 源点向每门课的第一个点连边，每门课的最后一个点连向汇点，权值为 ∞。 对于前置课程 (a,b)，只要从 a 的第 i 个点连向 b 的第 $i+1$ 个点，权值为 ∞。

时空复杂度	网络流复杂度，点数 $O(nm)$，边数 $O((n+k)m)$。

2.10

试题编号	NOV12 MARTARTS ★
试题名称	Martial Arts

题目大意	算法讨论
一个完全二分图，边有两个权值 $A[i][j], B[i][j]$。求一个完全匹配。令 $H = \sum A, G = \sum B$。 对手目的是最大化 $G - H$，其次最大化 G。你公布匹配方案之后，他可以删（也可以不删）一条匹配边，达到他的目的。 你的目的是最大化 $H - G$，其次最大化 H。求最优策略下的比赛结果。 $(n \leq 100)$	先不考虑第二关键字，只要最大化 $\sum V = \sum A - B$。那么一个匹配的答案实际上是总权值和减去最大的一条权值。我们可以从小到大枚举匹配中的最大边，将这条边强制选中，然后对比它小的边所形成的图做一次最大权匹配。一条边强制选中可以通过将边权改为 ∞，强制不选可以改成 $-\infty$。所以这个问题就变成了修改边权并动态维护最大完美匹配。 用 KM 算法，每次修改一条边 (u, v) 的边权时，先将 u 原先的匹配边断掉，然后重置 u 的顶标为 $A[u] = \max(w(u, j) - B[j])$，这样依然满足顶标的条件。然后再从 u 出发寻一条增广路。这样一次修改的代价只需要 $O(n^2)$ 考虑第二关键字的话，只要把参与运算的全部改为二元组即可。

时空复杂度	空间 $O(n^2)$，时间 $O(n^4)$。

2.11

试题编号	JUNE14 TWOCOMP
试题名称	Two Companies

题目大意	算法讨论
一棵树上有许多 A 路径和 B 路径，每个路径有权值。选出一些路径，使得每个点都不会同时被 A 路径和 B 路径经过。求最大权值总和。 $(n \leq 100000, 1 \leq m_1, m_2 \leq 700)$	这是一个二分图最大权独立集模型，可以转化为最小割。 构图时要对每一对 A, B 路径判一下是否相交，这只要对 LCA 讨论一下就好了。

时空复杂度	预处理 $O(n \log n + m_1 m_2)$。网络流点数 $O(m_1 + m_2)$，边数 $O(m_1 m_2)$。

2.12

试题编号	SEP12 PARADE ★
试题名称	Annual Parade

题目大意	算法讨论
一个有向图，边有非负权值。求若干条路径，使总代价最小。代价计算方式是： （1）每条路径的长度之和； （2）如果一条路径不是闭合的则对每条附加 C 的代价； （3）如果一个顶点没有被经过则每个点附加 C 的代价。 多次询问 C。 $(n \leq 250)$	关键在于分析这个代价计算出来的到底表示什么。首先对每一对点 (u, v) 都加上新边，权值为 $dis(u, v)$。则这个新图的答案不变，而且可以使得存在一个最优解中每个点都至多被访问一次（因为如果有多条路径挤在同个点上，可以把它移动到新边上去）。 这个条件满足之后，可以发现每新加入一条边，所花的总代价就会减少 C（要么成环，要么多经过一个点）。 于是只要对于边权二分图求出所有 k 匹配的最小代价和就好。可以用 KM 或费用流。

时空复杂度	空间 $O(n^2)$，时间 $O(n^3 + q \log n)$。

3 网络流和线性规划

3.1

试题编号	JUNE15 CHEFBOOK ★
试题名称	Chefbook

题目大意	算法讨论
给定 m 个整数对 (x,y),$1 \le x,y, \le n$ 和对应的 L_{xy}, S_{xy}, T_{xy}。求出 N 个非负整数 P_x 和 N 个非负整数 Q_x，使得 $W_{xy} = L_{xy} + P_x - Q_y$ 满足 $S_{xy} \le W_{xy} \le T_{xy}$，且 $\sum W_{xy}$ 尽量大。输出最大值或判无解，并输出方案。 $(n \le 100)$	首先写出线性规划的不等式组 $AP \le B$，其中 A 的每行有一个 $+1$ 和一个 -1，要最大化 $C^T P$。转化成对偶线性规划，需要最小化 $B^T Y$，满足 $A^T Y \le C$，此时 A 的每一列有一个 $+1$ 和一个 -1，将所有不等式相加会发现不等式组等价于 $A^T Y = C$，用这个等号作为流量平衡关系，可以画出一个最小费用最大流的网络模型。 然后要用对偶线性规划的解来得到原线性规划的解。这里要用到“互补松弛定理”，可以把原线性规划中的一些不等号加强为等号。然后用差分约束系统求出一组解，即可保证为最优解。

时空复杂度	空间 $O(n^2)$，时间 $O(costflow(n,m))$。

3.2

试题编号	FEB12 FLYDIST
试题名称	Flight Distance

题目大意	算法讨论
给定图的 n 个顶点和 m 对顶点间的距离。这些距离信息可能是有矛盾的（不满足三角不等式）。修改一个距离值需要 $\|d' - d\|$ 的代价（有理数）。求最小的总代价可以将信息修改成没有矛盾。 $(n \le 10, m \le 45)$	考虑线性规划。 对所有 $1 \le i < j \le n$ 建一个变量 $dis(i,j)$。 优化的式子中含有绝对值，这可以通过将两个变量作差去掉。对于每一条边 w_{ij}，建两个变量 d_1, d_2，表示边修改之后的距离为 $w_{ij} + d_1 - d_2$。那么要最小化的式子就是 $\sum d_1 + d_2$。 然后要限制 $w_{ij} + d_1 - d_2 = dis(i,j)$，以及 $w_{ij} + d_1 - d_2 + dis(j,k) \ge dis(i,k)$。 接下来跑单纯形就好了。

时空复杂度	单纯形变量数目 $O(n^2 + m)$，约数个数 $O(nm)$。

4 字符串

4.1

试题编号	SEPT13 TMP01 ★
试题名称	To Queue or not to Queue

题目大意	算法讨论
维护一个字符串，支持在开头加字符，在末尾删字符，并询问当前字符串的本质不同的子串数目。 ($q \le 1000000$)	如果能求出后缀数组，那么答案就是所有 height 值的和。 我们先离线搞出完整的串，并求后缀数组。然后就变成了每次插入一个后缀，或者当前所有后缀的末一个字符被删去。我们保证当前维护的所有后缀中，不存在两个后缀 a, b 使得 a 是 b 的前缀，如果出现了这种情况，就把 a 删去，而答案不会发生影响。这样子的话，每次删除末尾的字符，就相当于答案减少的值为当前的后缀数量。删了之后可能会出现新的上述不合法情况，要把它们删除掉。 具体的话可以用一个线段树，支持查询前驱后继，以及删除插入，并支持查询最小值。这里最小值指的是 $x.len - lcp(x, pre[x])$ 和 $x.len - lcp(x, suf[x])$ 的最小值，因为每次剔除不合法的都是从最小的开始删。

时空复杂度	空间 $O(n)$，时间 $O(n \log n)$。

4.2

试题编号	APRIL12 TSUBSTR
试题名称	Substrings on a Tree

题目大意	算法讨论
一棵有根树，结点标了字母。子串定义为从一个结点走到孩子的路径上形成的串。求本质不同的子串有几个。并且每次给出一个新的字母表顺序，输出此意义下的字典序第 k 小字符串。字母随机生成。 ($1 \le n \le 250000, q \le 50000$)	这是一个 SAM 裸题。对这棵树建 SAM，然后算出 SAM 上每个结点可以走到多少个字符串，这可以按结点的 len 升序 DP 求出。然后每次从根结点开始在 SAM 上跑就可以了。

时空复杂度	空间 $O(n\vert\Sigma\vert)$，时间 $O((n + \sum \vert ans\vert)\vert\Sigma\vert)$。

4.3

试题编号	AUG13 LYRC
试题名称	Music & Lyrics

题目大意	算法讨论
给出 w 个单词和一篇文章，问每个单词在文章中出现了多少次。 ($w \le 500$, 每个单词长度 $p \le 5000$，文章长度 $s \le 5000000$，字符集大小 $\Sigma = 63$)	经典问题。对 w 个单词建 AC 自动机。然后将文章在自动机上跑一遍，并记录一下每个结点被经过了几次。 一个单词的出现次数就是对应结点在 fail 树上的子树和。

时空复杂度	空间 $O(wp\vert\Sigma\vert)$，时间 $O(wp\vert\Sigma\vert + s)$。

4.4

试题编号	DEC12 DIFTRIP
试题名称	Different Trips

题目大意	算法讨论
一棵有根树，u 上标的字符为 $deg[u]$。子串定义为从某个结点到根的路径的前缀。求有多少个本质不同的子串。 $(n \leq 100000)$	链上的做法就是求出后缀数组，然后用总子串数目减去 height 数组的和。 树上的情况也可以用后缀数组的做法。把 u 到根结点的路径当做后缀，也是类似的倍增，然后求出 height 数组。但是这里需要注意可能存在完全相同的后缀，需要处理一下。可以在倍增及算 height 的过程中加特判，也可以先把树上的等价点合并变成一个 trie 再做。 另外本题也可以用 SAM 做，但由于字符集较大，需要用 map 存边。

时空复杂度	空间 $O(n\log n)$，时间 $O(n\log n)$。

4.5

试题编号	APRIL13 STRQUERY
试题名称	String Query

题目大意	算法讨论
维护一个字符串 s，支持在左、右、正中间插入或者删除一个字符，询问给定串在 s 中出现了几次。 $(q \leq 150000, \sum \|str\| \leq 1500000)$	用后缀平衡树可以支持在一边插入或删除，那么把两个后缀平衡树拼起来就可以支持两边操作了，再把两个这样的结构拼起来就支持在中间操作了。 询问的话，完全包含在一个结构内的可以在平衡树上二分查找；跨越两个结构的可以用 KMP。

时空复杂度	空间 $O(n)$，时间 $O(\|s\|\log n)$。

5 数论

5.1

试题编号	SEP11 SHORT
试题名称	Short

题目大意	算法讨论
给定 n,k，求出有几对整数 (a,b) 满足 $n<a<k,n<b<k,(a-n)(b-n)\|ab-n$。 $(0\le n\le 100000,n<k\le 10^{18})$	先特判掉 $n=0$ 的情况。 令 $p(a-n)(b-n)=ab-n$，得 $b=n+\frac{n(a-1)}{p(a-n)-a}$。 考虑 $a\le b$ 的情况并枚举 a，根据题中关系可以化出一个二次不等式从而得到 a 的上界 $a\le 3.42n$，然后只要枚举 $n(a-1)$ 的所有约数 $d=p(a-n)-a$，然后看 p 是否为整数即可。为了枚举约数，可以先筛出 $3.42n$ 以内所有数的质因数分解，然后就能得到 $n(a-1)$ 的质因数分解，于是 DFS 就可以了。 对于 a 较大的情况，可以改成枚举上式中的 p。由一些不等关系可以得到 $p\le\frac{a^2-n}{(a-n)^2}$，这个范围会比较小，于是可以加快速度。

时空复杂度	空间 $O(n)$，时间玄学。

5.2

试题编号	DEC11 SHORT2
试题名称	Short II

题目大意	算法讨论
给定质数 p，有多少组 (a,b) 满足 $a,b>p,(a-p)(b-p)\|ab$。 $(p<10^{12})$	可以转化成求有多少组 $a\ge 1,b\ge 1$ 使得 $ab\|p(a+b+p)$。 当 $p\|a,p\|b$ 时可以证明有 5 组解。 当 $p\nmid a,b=b'p$ 时，令 $k=ab'/(a+pb'+p),d=ka-p$，则有 $d\|a+p,a\|d+p$，且 a,d 均与 p 互质，此时有 $ad\|(a+p)(d+p)$，和原题的式子相同，可以转化成第三种情况。 所以现在只要考虑 $p\nmid a,p\nmid b$，即有 $a+b+p=kab$。$a=b$ 时有 $a=b=1$，只要考虑 $a<b$。那么 $(a-1)(b-1)\le p+1$，则 $a<w=1+\sqrt{p+1}$。令 $d=ka-1=(a+p)/b$。当 d 固定时，由 $d\|a+p,a\le d+1$ 可知只有不超过 2 种可能的 a，所以先枚举 $d\le\sqrt{p+w}$。当 $d>\sqrt{p+w}$，有 $b=(a+p)/d<(a+p)/\sqrt{p+w}<\sqrt{p+w}$，所以枚举 b，此时也可以唯一的确定 a 和 d。

时空复杂度	空间 $O(1)$，时间 $O(\sqrt{p})$。

5.3

试题编号	NOV11 LUCKYDAY ★
试题名称	Lucky Days

题目大意	算法讨论
一个数列 S 如下定义： $S[1] = A$ $S[2] = B$ $S[i] = (X \cdot S[i-1] + Y \cdot S[i-2] + Z) \bmod P, i \geq 3$ 对于给定的 C，求出区间 $[l, r]$ 中的 i 有多少个 $S[i] = C$。 t 组询问每组给定 l, r。 $(p \leq 10007, t \leq 20000, l_i, r_i \leq 10^{18})$	首先特判掉退化情形 $(y = 0)$，这时循环节长度是 $O(P)$ 的，直接暴力弄出循环节就好。 接下来考虑 $y \neq 0$。写成转移矩阵的形式 $M^k(B, A, 1)^T = (S[k+2], S[k+1], 1)^T$。 这显然是一个纯循环数列。首先求出循环节，这里要用小步大步，即把 k 写成 $k = iD - j$ 的形式。循环节长度是 $O(p^2)$ 的。 然后要找出所有 C 出现的位置，这里设 $S[k+2] = C$，然后要枚举 $S[k+1]$ 的值 $\in [0, p-1]$。然后也要使用小步大步。 这样要在哈希表中进行 p^2/D 次插入和 pD 次查询。取 $D = O(\sqrt{p})$ 可以达到最优复杂度。

时空复杂度	空间 $O(p^{3/2})$，时间 $O(p^{3/2})$。

5.4

试题编号	AUG11 DIVISORS
试题名称	Something About Divisors

题目大意	算法讨论
对于给定的正整数 b 和 x，求满足条件的正整数 n 的个数: 要求对于 n，至少存在一个数 $d(n < d \leq b)$ 能整除 nx。 $(testcases \leq 40, b \leq 10^{12}, x \leq 60)$	令 $k = nx/d$，有 $k < x$，我们固定 k，然后统计 n 的数量，为了去重，我们只统计不存在 $k < j < x$ 且 $j \mid nx$ 的 n。 $k \mid nx$ 可以写成 $k/\gcd(k, x) \mid n$，于是可以写成 $n = m \cdot k/\gcd(k, x)$ 从而统计对应的 m 的数量。根据题目约束可以得到一个 m 的上界 L。 然后要把存在 $j \mid n$ 的 m 给去掉。$j \mid n$ 等价于 $D(j) \mid m$，其中 D 是一个仅关于 j, k, x 的式子。我们对于 $k < j < x$ 的所有 j，将 $[1, L]$ 中 $D(j)$ 的倍数从答案里去掉。这里要用容斥原理。 然后需要用各种优化才能通过，首先将存在 $D(j_1) \mid D(j_2)$ 的 $D(j_2)$ 给去掉。然后在枚举下一个 $D(j)$ 进行扩展的时候，可以写成归并有序队列的形式。扩展过程中遇到数量为 0 的，或者当前 lcm 已经大于 L 的，就直接删去。

时空复杂度	空间玄学，时间玄学。

5.5

试题编号	OCT13 FN
试题名称	Fibonacci Number

题目大意	算法讨论
斐波那契数列 f_n，求出一个最小的 n 使得 $f_n \equiv c(\text{mod} p)$。 ($p \leq 2 \times 10^9$，保证 5 是 p 的二次剩余)	斐波那契数列的通项公式为 $f_n = (\phi^n - (-\phi)^{-n})/\sqrt{5}$，其中 $\phi = (1+\sqrt{5})/2$。 首先枚举 n 的奇偶性，然后得到一个关于 ϕ^n 的二次方程，求出 ϕ^n 再用小步大步求出 n 即可。这个过程中需要求模意义下的平方根，可以使用 Cipolla 算法。

时空复杂度	空间 $O(\sqrt{p})$，时间 $O(\sqrt{p})$。

5.6

试题编号	JAN12 MISINT2
试题名称	Misinterpretation 2

题目大意	算法讨论
有多少个长度在 $[L, R]$ 的小写字母字符串，满足：把它的所有偶数位移到前面，奇数位接在后面，不改变原来的相对顺序，得到的字符串是不变的？ $R - L \leq 50000, R \leq 10^{10}$	固定长度 m，那么答案显然为 $26^{g(m)}$，$g(m)$ 是置换 p 的轮换数目。可以看出 $g(2k+1) = g(2k)+1$，那么只要考虑 $m = 2k$ 的情况。容易发现 $p(x) = 2x \bmod (2k+1)$。于是可以证明 $g(m) = \sum_{d\|(m+1),d>1} \phi(d)/\text{ord}_d(2)$。接下来只要对 $[L, R]$ 内的奇数快速求出这个即可。 先筛出 $\sqrt{R}$ 以内的质数，然后用筛法可以得到 $[L, R]$ 内的所有质因数分解，于是可以 DFS 得到所有约数。现在要求 ord，首先由 p, q 互质时 $\text{ord}_{pq}(a) = \text{lcm}(\text{ord}_p(a), \text{ord}_q(a))$，于是只需要考虑模 p^k 即可。将 $\phi(p^k)$ 分解质因数，然后由 $X = \phi(p^k)$ 开始不断除以某个质因数直到不满足 $2^X \equiv 1$ 时停止，然后再对下一种质因数继续除。最后剩下的 X 就是所求的阶。

时空复杂度	空间 $O(R-L+\sqrt{R})$，时间 $O((\sqrt{R}+R-L)\log\log\sqrt{R}\log R+(R-L)\log P)$。

6 代数，FFT

6.1

试题编号	AUG15 CLOWAY
试题名称	Future of draughts

题目大意	算法讨论
给 T 张无向图，每个点数 n_i。 每次询问：只考虑编号在 $[L,R]$ 内的图，一开始在每个图选定一个出发点，然后每次选择一些图（不能不选），把里面的点沿着边动一下。如果所有图里的点都回到出发点那么游戏可以结束。求在 k 次之内结束游戏的方案数。对 10^9+7 取模。 $(n,T\leq 50, k\leq 10^4, Q\leq 2\times 10^5)$	首先将每个图表示成邻接矩阵。题中要求若干步后回到出发点，它的方案数对应的是矩阵的 trail。于是第一步需要求出每个矩阵 A 的若干次幂 A^k 的 trail： 对于 $k\leq n$，暴力求。对于 k 较大的情况，可以递推。根据 Caylay-Hamilton 定理可知道递推式是矩阵的特征多项式。因此只需要求出特征多项式即可，一个方便的做法是给 x 代入几个值算出行列式，然后用 Lagrange 插值。 然后按照题意容斥，容斥的式子是一个类似二项式反演的东西，把组合数的分子分母拆出来就可以化成卷积的形式，然后用 FFT 即可。 这里的模数不太舒服，可以先用 3 个支持 FFT 的模求出答案，然后用中国剩余定理合并。用正确的合并姿势可以不爆 long long 也不需要浮点数。

时空复杂度	空间 $O(T^2k)$，时间 $O(T^2k\log k+Tn^4)$。

6.2

试题编号	FEB15 CUSTPRIM
试题名称	Payton numbers

题目大意

定义三元组 (a,b,c) 的乘法，其中 $c=11$ 或 24：

```
def multiply((a1,b1,c1), (a2,b2,c2)):
    s = (a1a2 + b1b2 + c1c2) +
(a1b2 + b1a2) + (c1 + c2)
    t = floor[s/2] + 16(c1 + c2) - c1c2
    A = (t - 2(a1b2 + b1a2) -
(a1c2 + c1a2) + 33(a1 + a2)
          + (b1b2 - a1a2))
    B = (t - 5(a1b2 + b1a2) -
(c1b2 + b1c2) + 33(b1 + b2)
          + (2b1b2 + 4a1a2))
    if s is even:
        return (A-540,B-540,24)
    else:
        return (A-533,B-533,11)
```

给定一个三元组，求它是否为素数。
$(a,b\leq 10^7, T\leq 10^4)$

算法讨论

用一些特殊的手段可以得到如下结论：
定义 $\phi(a,b,c)=(33-2a-c)+(b-a)\omega$，其中 $\omega=(1+\sqrt{-11})/2$。
定义 $Nx=N(a+b\omega)=a^2+ab+3b^2$。
则当 x 不是整数时，x 是质数当且仅当 Nx 是质数。
当 x 是整数时，x 是质数当且仅当 x 质数且要么 $|x|=2$ 要么 $|x|\neq 11$ 且 $-11 \bmod x$ 没有二次剩余。
接下来只需要对普通的整数进行质数判定就可以了，这可以用 miller-rabin 算法。

时空复杂度	空间 $O(1)$，时间 $O(T\log a)$。

6.3

试题编号	MARCH15 RNG
试题名称	Random Number Generator

题目大意	算法讨论
对于一个 k 阶线性齐次递推数列，求出第 n 项的值。模数可以 FFT。 $(k \le 30000, n \le 10^{18})$	这是一个经典问题。 设初始值向量 X，转移矩阵为 A，我们要求 $A^n X$ 的最末一行的值。可以证明特征方程 $f(x)$ 满足 $f(A)=0$。于是可以用多项式除法将 A^n 表为 $A^{k-1}, \cdots A^0$ 的线性表示。 由于 n 较大需要一边倍增一边除。

时空复杂度	空间 $O(n)$，时间 $O(k \log k \log n)$。

6.4

试题编号	AUG14 SIGFIB
试题名称	Team Sigma and Fibonacci

题目大意	算法讨论
求 $\sum_{x+y+z=N} 6xyzf_xf_yf_z \bmod m$。$f_i$ 是斐波那契数。 $n \le 10^{18}, \sum m \le 10^6$	根据斐波那契数列的生成函数，容易推出所求答案为 $$[x^N]\frac{6x^3(1+x^2)^3}{(1-x-x^2)^6}$$ 它的系数可以写成一个 12 阶齐次线性递推数列。可以用经典的 $O(k^2 \log n)$ 做法来求（这里 $k=12$，只要用暴力乘除法即可）。 但这样跑不出 5×10^5 组数据。解决方法是对 $m \le 200$ 左右的预处理答案，因为循环节很短。

时空复杂度	空间 $O(1)$，时间 $O(\log n)$。

6.5

试题编号	MARCH13 CHANGE ★
试题名称	Making Change

题目大意	算法讨论
n 种硬币，第 i 种面值 d_i，硬币无限多。要凑出 C 的总和总共有几种方法模 10^9+7。d_i 两两互质。 $(n \le 50, c \le 10^{100}, d_i \le 500)$	较复杂的一题。首先将答案的生成函数部分分式分解得到 $$\prod \frac{1}{1-x^d} = \frac{A(x)}{(1-x)^n} + \sum \frac{B_d(x)}{1+\cdots+x^{d-1}}$$ 为求出 $B_d(x)$，将两边同乘 $(1+\cdots+x^{d-1})$，并用 $d-1$ 个单位根代入。一个结论是 $\frac{1}{1-\omega_d}$ 可以写成 ω_d^i 的多项式。所以可以变成 n 个多项式的卷积，且这个卷积能根据它的特殊形状快速计算。 然后 $[x^i]\frac{A(x)}{(1-x)^n}$ 是 i 的 $n-1$ 次多项式。只要暴力前 n 项就可以用拉格朗日公式算出第 C 项。

时空复杂度	空间 $O(nd)$，时间为 $O(n^2d)$

6.6

试题编号	DEC13 REALSET ★
试题名称	Petya and Sequence

题目大意	算法讨论
输入数组 $a_0, \cdots, a_{n-1}$，判断它不断右移所形成的循环矩阵是否是非奇异矩阵。 $(n \le 30000)$	令 $f(x) = \sum_{0 \le i \le n-1} a_i x^i$，由维基可得循环矩阵的秩等于 $n - \deg \gcd(f(x), x^n - 1)$，所以只需判断 $f(x)$ 是否被某个分圆多项式 $\Phi_d(x)$ 整除。 可以证明这等价于 $x^d - 1 \mid f(x) \prod_{p \mid d} (x^{d/p} - 1)$，因此只需要暴力相乘并取模即可，由于乘和除的式子都只有两项，所以一次操作复杂度是线性的。

时空复杂度	空间 $O(n)$，时间 $O(nd(n))$。

6.7

试题编号	JAN13 CUCUMBER ★
试题名称	Cucumber Boy and Cucumber Girl

题目大意	算法讨论
有 B 个 $n \times n$ 矩阵 $Q_1, \cdots, Q_B$。 对于每对 $(a, b), 1 \le a < b \le B$，新矩阵 $C_{a,b}$ 满足 $\displaystyle C_{a,b}[i][j] = \sum_{1 \le k \le n} Q_a[i][k] Q_b[j][k]$ 若有奇数个 n 排列 p，使得 $C_{a,b}[i][p_i]$ 中至少有一个奇元素，那么 (a, b) 是好数对。 求好数对个数。 $(n \le 60, b \le 8000)$	首先在矩阵的最后一列添上一列 1，得到 b 个 $n \times (n+1)$ 矩阵 A_i。所需要求行列式的矩阵就是 $A_a A_b^T$。 按照 Cauchy-Binet 将它的行列式写成 $n+1$ 个矩阵的积的行列式之和，每个矩阵都是原来的矩阵去掉对应的那一列（如果是转置就是去掉那一行）。于是只要对每个矩阵预处理出去掉任一列后的行列式即可。 求行列式的话需要用高斯消元。这里是模 2 意义下，所以等价于判断是否满秩。每去掉一列都重新消元的做法太慢，一个快的做法是先对整个大的 $n \times (n+1)$ 矩阵消元，这样会多出不存在主元的某一列，然后稍微判一下就能知道去掉某一列后是否满秩了。 因为是在模 2 意义下运算，可用位运算加速。

时空复杂度	空间 $O(Bn^2)$，时间 $O(\frac{1}{64}(B^2 n + Bn^3))$。

6.8

试题编号	OCT11 PARSIN
试题名称	Sine Partition Function

题目大意	算法讨论
求值： $\displaystyle \sum_{k_1 + k_2 + \cdots + k_m = n} \sin(k_1 x) \cdots \sin(k_m x)$ $(t \le 10, m \le 30, n \le 10^9)$	令 $f(n, m)$ 表示答案，而 $g(n, m)$ 表示将原题求和式中最后一个 sin 换成 cos 后的答案。 那么根据一些三角恒等关系可以得到 $g(0, 1) = 1, g(n, m) = f(n, m-1) + \cos x g(n-1, m) - \sin x f(n-1, m), f(n, m) = f(n-1, m) \cos x + g(n-1, m) \sin x$。 然后用矩阵乘法优化这个 dp 即可。

时空复杂度	空间 $O(m^2)$，时间 $O(m^3 \log n)$。

6.9

试题编号	JULY11 YALOP
试题名称	Trial of Doom

题目大意	算法讨论
一个 $n \times m$ 的网格，每个格子一盏灯。从 $(1,1)$ 走到 (n,m)，每走一步，目标格子的灯以及它的四周的灯会改变状态。现在有 k 个亮灯，问能否全部熄灭。 $(1 \le n \le 40, 1 \le m \le 10^9, 1 \le k \le 10000)$	首先当 $n, m \le 2$ 时，注意到在一个 2×2 的方格中我们可以从一个顶点走到另一个而不改变灯的状态。所以我们可以忽略路径的限制，随意的按开关。 假如第一行的方案以确定，就可以从上往下贪心的按灯，如果最后一行能消光则是合法的。我们可以算出中间的每个亮灯对于最后一行的贡献，因为每行的长度很小，所以这个东西是会很快循环的。而且每个灯对最后一行的贡献都是独立的，可以叠加。然后再考虑第一行的方案，使得最后一行消完，这就可以用高斯消元解方程判断是否有解了。 如果是 $n = 1$ 的情况，则需要考虑路径的限制。注意到可以以 2 为步长不改变状态地随意跳，所以稍微分析一下奇偶性就可以了。

时空复杂度	空间 $O(n \cdot statenum)$，时间 $O(n \cdot statenum + k + n^2)$。

6.10

试题编号	JUNE11 CLONES
试题名称	Attack of the Clones

题目大意	算法讨论
考虑所有 $f : \{0,1\}^n \to \{0,1\}$ 的布尔函数的集合。 集合 Z 是 0-preserving 的，集合 P 是 1-preserving 的，集合 D 是 self-dual 的，集合 A 是 affine 函数。 给定一个包含 Z, P, D, A 和 $\cup, \cap, \setminus, \complement$ 的表达式。求出得到的集合的元素个数。 $(n, length \le 100)$	首先对于 2^4 种情况（Z, P, D, A 分别是包含它还是不包含），分别算出这个集合中有多少个元素。这需要根据定义略加分析就能算出。然后就相当于一个 16 位二进制数之间的运算了。 剩下的就是一个经典的表达式求值问题了。做法是开一个数栈和一个运算符栈，遇到括号的时候注意调整优先级。还要注意一下取补集运算是单目右结合的。

时空复杂度	空间 $O(length)$，时间 $O(length + \log n)$。

6.11

试题编号	AUG13 PRIMEDST
试题名称	Prime Distance On Tree

题目大意	算法讨论
从树上均匀随机选出两个结点，它们之间的距离是质数的概率是多少。 $(n \le 50000)$	只要统计出树上每种长度的路径有多少条即可，这是一个经典问题。 点分治，在合并子树的信息时，发现其实是卷积的形式，于是用 FFT 统计答案即可。 为了保证复杂度，需要按子树的最大深度从小到大处理。

时空复杂度	空间 $O(n)$，时间 $O(n \log^2 n)$。

6.12

试题编号	NOV12 COUNTARI
试题名称	Arithmetic Progressions
题目大意	算法讨论
一个整数数列 $a_1, \cdots, a_n$。有多少组 $1 \le i < j < k \le n$ 使得 $a_j - a_i = a_k - a_j$。 $n \le 100000, 1 \le a_i \le 30000$	对数列按下标分块。首先考虑 i, j, k 不在同一块的情况，枚举 j 在哪一块，然后将它左边的块的权值数组与它右边的用 FFT 求一下卷积，即可统计。 对于 i, j 在同一块的情况，直接暴力枚举 i, j，可求出对应的 a_k 的值。所以维护一个 cnt 数组记录一下目前每个数出现了几次即可。
时空复杂度	空间 $O(n + a)$，时间 $O(n\sqrt{a} \log a)$。

6.13

试题编号	FEB15 DEVLOCK
试题名称	Devu and Locks
题目大意	算法讨论
多少个 n 位数码，每位为 0 到 9 的数字（允许前导 0），使得各位数码之和不超过 m，且整个数被 p 整除。 对于 $0 \le m \le M$ 均输出答案。 (最大的数据中 $n \le 10^9, p \le 16, M \le 15000$)	把 n 个位置按 $10^i \bmod p$ 分成 p 组（用找循环节的方法）。对于第 i 组定义生成函数 $f_i(x)$，其中第 k 项系数表示这些位置中填入 k 的总和有多少种方案，它是 $(1 + x + \cdots + x^9)$ 的若干次幂。维护当前答案为 p 个多项式 $g_i(x)$，第 i 个的 j 次项系数表示 j 的数码和凑出 i 的余数的方案数。每次把 $f_i(x)$ 更新到答案里，具体方法是把 $f_i(x)$ 也类似地拆成 p 个多项式 $h_j(x)$，然后和 $g_j(x)$ 两两相乘再加起来（多项式的卷积），但这样太慢，更快的方法是把这些多项式都 DFT，再对点值两两相乘相加，再逆 DFT 回去。
时空复杂度	空间 $O(pm)$，时间为 $O(pm \log n \log m + p^2 m \log m + p^3 m)$。

6.14

试题编号	JULY15 EASYEX
试题名称	Easy exam
题目大意	算法讨论
一个面上写着 $1 \sim K$ 的骰子，掷 N 次后，令 a_i 为点数 i 的出现次数。求出 $a_1^F \times \cdots \times a_L^F$ 的期望。（模质数意义下） $(0 < N, K \le 10^9, 0 < F \le 1000, 0 < L \times F \le 50000)$	如果记 b_i 为第 i 次掷骰子得到的点数。那我们相当于从 b 数组中任抽取出一个有序 $L \times F$ 元组 $(b_{i_1}, \cdots, b_{i_{L \times F}})$，考虑它满足 $b_{i_1} = \cdots = b_{i_F} = 1, b_{i_{F+1}} = \cdots = b_{i_{2F}} = 2, \cdots$ 的概率是多少，我们对每个有序 $L \times F$ 元组将这个概率求和，就是答案。 显然若 b_i 不同则 i 不同，于是可以分成 L 个不交的下标集合。考虑一下每个集合的大小是 $i (1 \le i \le F)$ 时有多少种可能。然后就可以 DP 了，这个 DP 可以转化成求多项式的幂的形式，需要用 FFT 加速。
时空复杂度	空间 $O(LF + F^2)$，时间 $O(P^2 + F^2 + LF \log(LF))$，$P$ 是模数。

6.15

试题编号	DEC12 QPOLYSUM
试题名称	Quasi-Polynomial Sum

题目大意	算法讨论
给定一个多项式 $P(X) = C_D X^D + \cdots + C_1 X + C_0$，其中 $C_0, C_1, \cdots, C_D$ 是整数。 给定非负整数 Q，正整数 M, N，求 $(P(0)Q^0 + P(1)Q^1 + \cdots + P(N-1)Q^{N-1}) \bmod M$ 的值。 本题不直接给你 $C_0, C_1, \cdots, C_D$，而只会告诉你 $A_0, A_1, \cdots, A_D$, 其中 $A_i = P(i) \bmod M$。 $1 < M < 10^{18}, 1 \le N < 10^{100000}, 0 \le D < 20000$，$M$ 不能被 2 至 $D+14$ 中的任意一个数整除。	首先 $Q = 0$ 直接输出。 $Q = 1$ 时，就是一个前缀和，是一个 $D+1$ 次多项式，插值就好。 $Q \ne 0, 1$ 时，可以归纳证明存在一个 D 次多项式 F 使得要求的 sum 等于 $F(N)Q^N - F(0)$。 只要求出 $F(0)$ 到 $F(D)$ 的值，就可以插值求出答案了。这些值之间是满足递推关系的，然后用 $D+1$ 阶差分为零的条件可以列出一个一元一次方程，求出来即可。 上面的过程要求存在 Q 和 $1-Q$ 的逆元，所以要把 M 中与它们有公因子的部分提出去，变成三个模数，分别算然后用 CRT 合并起来。互质的模用上述做法。Q 对应的模直接暴力。$Q-1$ 对应的模，用二项式展开一下，也是可以做的。

时空复杂度	空间 $O(D + \log N)$，时间 $O(D + \log N)$。

7 组合计数，DP

7.1

试题编号	JULY14 SEAEQ
试题名称	Sereja and Equality

题目大意	算法讨论
两个长度为 n 的数组 A,B 相似，如果对于所有 $i(1 \le i \le n)$，满足 $C(A,A_i)=C(B,B_i)$。其中 $C(X,x)$ 等于满足 $X[j]<x(1\le j\le n)$ 的 j 的数目。 对于两个排列 P_1,P_2，定义函数 $F(P_1,P_2)$ 等于满足 $P_1[l...r]$ 相似于 $P_2[l...r](1\le l\le r\le n)$ 并且 $P_1[l...r]$ 包含不超过 E 个逆序对的数对 (l,r) 的数目。 对 P_1,P_2 取遍所有 n 个元素的排列 $F(P_1,P_2)$ 的总和是多少? $(n\le 500, t\le 10000)$	首先用一个经典的 $O(n^3)$ 的 DP 求出 $f[i][j]$ 表示 i 排列中逆序对数量不超过 j 的有几个，转移只要考虑在第 $i+1$ 位插入的数在前 i 个数中的排名是多少就好了。 然后只要枚举每个长度 l 算出对答案的贡献，首先根据 DP 得到的排列数量，然后乘上 $\binom{n}{l}$，再乘上剩下数字的排列数 $(n-l)!$。对两个排列都这样乘一遍。

时空复杂度	空间 $O(n^3)$，时间 $O(n^3+Tn)$。

7.2

试题编号	JUNE13 SPMATRIX
试题名称	Count Special Matrices

题目大意	算法讨论
设 $n\ge 3$ 是一个固定正整数。 令 $a[1..n][1..n]$ 是一个 $n\times n$ 的整数矩阵。矩阵满足条件： $(1)a[x][y]=0$ $(2)a[x][y]=a[y][x]>0$ $(3)a[x][y]\le \max(a[x][z],a[z][y])$ $(4)a[x][y]\in\{1,2,\cdots,n-2\}$ $(5)\forall\ k\in\{1,2,\cdots,n-2\}$, $\exists x,y\, s.t.\, a[x][y]=k$ 模 10^9+7 $(n\le 10^7, t\le 10^5)$	用奇怪的方法可以得到答案的表达式为 $\frac{n!(n-1)!}{3\times 2^{n-1}}\left(3n/2-2-\sum_{i=1}^{n-1} i^{-1}\right)$ 用 $O(n)$ 时间预处理即可。 为了减少常数，分母的 2^{n-1} 留到每次询问再除。右边的调和级数不要用逆元求，而是和 $(n-1)!$ 乘起来一起预处理。

时空复杂度	空间 $O(n)$，时间 $O(n+t\log n)$。

7.3

试题编号	JAN14 CNTDSETS
试题名称	Counting D-sets

题目大意	算法讨论
考虑整点，一个点集的直径定义为两点中某一维的差的绝对值的最大值的最大值。可以平移得到的点集视作同一个。求有多少个 n 维点集的直径是 d。 $(n\le 1000, d\le 10^9)$	转化为计数直径不超过 d 的点集数量。然后用 d 的答案减去 $d-1$ 的答案即可。 为了去重，考虑将点集平移使得每一维坐标非负，且每一维都至少有一个点的该维坐标为 0，那么所有点都落在 $[0,d]^n$ 的区域里。为了限制每一维都有取到 0 的点，可以用容斥原理。

时空复杂度	空间 $O(n^2)$，时间 $O(n^2+tn\log P)$。

7.4

试题编号	JULY11 BB ★
试题名称	Billboards

题目大意	算法讨论
长度为 n 的 01 序列，每连续 m 位中至少有 k 个 1，求有多少种方案使得 1 的数量最少。 $(1 \le k \le m \le 50, m \le n \le 10^9,)$	将序列分成 $\lfloor n/m \rfloor + 1$ 段，前几段长度为 m，最后一段长 $n \bmod m$。可以发现，达到最优的情况下，前面每段里恰好有 k 个 1，最后一段恰有 $\max(0, k-m+(n \bmod m))$ 个 1。按照最后一段是否有 1，可以分成两种情况。再把序列倒过来分段，使得第一段长度是 $n \bmod m$，后面接着 $\lfloor n/m \rfloor$ 段，也可以得到同样的结论，然后和前一种分段结合起来看的话，画一画可以发现序列分成好多个小段，每隔一小段就有一小段是全 0（或全 1，看刚才分成哪一类）的，所以只要考虑剩下小段里怎么填的，假设这些小段都是长为 p，里面要填 q 个 1。把每小段中填 1 的下标写下来，可以组成一张表，容易证明表是按行严格递增，按列不严格递减的。 然后根据杨氏图表的相关结论即可得到结果。

时空复杂度	空间 $O(1)$，时间 $O(km)$。

7.5

试题编号	MARCH13 LECOINS ★
试题名称	Little Elephant and Colored Coins

题目大意	算法讨论
n 个硬币，第 i 个面值 v_i，颜色为 c_i。q 次询问，要凑出恰好 n 的面值，用到的硬币最多能有几种颜色？ $(n \le 30, v_i \le 200000, q \le 100000, n \le 10^{18})$	取其中一个 v_i 作为模 m，令 $f[i][j]$ 表示已经用到了 i 种颜色，已凑成的面值模 m 的余数是 j 时的最小面值。这样的话只要这个面值小于等于 n，就可以用若干个 m 面值的硬币凑出 n。 这个 dp 转移的形式是一个最短路，但其实不需要使用最短路算法，因为转移边是形成若干个环的，只要沿着环暴力更新就可以了。

时空复杂度	空间 $O(n \min(v_i))$，时间 $O(n^2 \min(v_i) + nq)$。

7.6

试题编号	MAY12 LEBOXES
试题名称	Little Elephant and Boxes

题目大意	算法讨论
n 个宝箱，第 i 个里有 p_i 的概率装有 v_i 元钱，有 $1-p_i$ 的概率装有一个钻石。 m 个物品，第 i 个需要花费 c_i 元钱加上 d_i 颗钻石。求打开所有宝箱后期望最多能买多少个物品。 $(n, m \le 30, v_i, c_i \le 10^7, d_i \le 30)$	首先容易用简单的 dp 求出 $f[i][j]$，表示手里有至多 j 个钻石时，想买 i 个物品最少花几元钱。 然后考虑 meet in middle，处理出前 $n/2$ 个宝箱的所有 $2^{n/2}$ 种可能性，以及后 $n/2$ 个宝箱的所有可能性。枚举一下开宝箱恰好得到的钻石数量 j，然后枚举想买的物品数 i，然后看看两边各取一个相加的结果 $\ge f[i][j]$ 的概率，这可以通过排序后维护各维护一个指针，线性扫一遍来得到。

时空复杂度	空间 $O(2^{n/2} + mn)$，时间 $O(nm2^{n/2})$。

7.7

试题编号	DEC11 HYPER
试题名称	Hypertrees

题目大意	算法讨论
超图是每条边连接三个不同顶点的图。 超树是去掉任一条边就会不连通的超图。 求 n 个点的，点有标号的超树的数量。 $(n \le 17)$	一个超树可以被划分成若干个点双连通分量。先考虑统计点双连通超图的数目： 3 个点的只有一种情况。$n \ge 4$ 时，可以发现，对于图中的任意一个边，它连接的三个点中恰有一个点的度数是 1。于是可以把这个度数为 1 的点删去，那么这个超图变成了一个普通的点双连通图，它的点数 + 边数即为原超图的点数。暴搜点数 + 边数 ≤ 17 的点双连通图数量，即可得到答案。搜的时候先不考虑重边，然后再用一些计数技巧算出有重边时的数目。 然后考虑把若干个点双连通图拼成一棵超树，这可以记忆化暴搜。先固定图中的一个点双连通分量 B 作为“根”，然后搜索 B 中的结点分别拥有的子树（这里要记忆化）。最后要除以这个图的双连通分量数目，因为每一个都被当做了一次根。

时空复杂度	空间 $O(n^2)$，时间玄学。

7.8

试题编号	SEP11 CNTHEX
试题名称	Counting Hexagons

题目大意	算法讨论
选出 6 根棒子，长度为 $[1,n]$ 的整数，拼出一个面积为正的六边形。要求最长边长度至少是 L，其它边的长度不超过 X，同种长度的棒子不超过 K 根。两个六边形是不同的当且仅当边长集合不同。求能拼出几种六边形。 $(2 \le n \le 10^9, 2 \le l \le n, n-l \le 100, 1 \le x < l, 1 \le k \le 5)$	拼出面积为正的六边形的条件是其他边的长度之和严格大于最长边长度。可以补集转化统计其他边长之和不超过最长边的。 首先枚举最长边的长度。 然后容斥相等关系，可以发现只有以下几种组合：11111,1112,113,122,14,23,5。 接下来容斥边长是否大于 X，如果设定为大于就将边长替换为 $len - X$ 为正整数，然后将上界也相应减少 X。最后能得到一个非负整数的不定方程 $x_0 + \sum k_i x_i = l$，其中 $\sum k_i = 5$，x_0 是将不等号转为等号而添加的变量。对于每种系数组合，都可以求出其解数关于 l 的通项公式，直接代入求得答案即可。

时空复杂度	空间 $O(1)$，时间 $O((n-l) \times 7 \times 2^5 \times \log p)$。

7.9

试题编号	SEP12 KNGHTMOV
试题名称	Knight Moving

题目大意	算法讨论
从 $(0,0)$ 走到 (x,y)，每次走的位移向量可以是 (x_a,y_a) 或 (x_b,y_b)。 有 k 个位置是禁止进入的。 求总方案数，要判无解或无穷个解的情形。 ($k \le 15$，输入坐标的绝对值 $A \le 500$)	当两个向量不共线时，则走到任意一个点所需的 a 向量和 b 向量数量是可以解出来的，方案数即为 $\binom{n_a+n_b}{n_b}$。现在有 k 个禁止位置，由于 k 很小，只要容斥就可以了。 如果共线，就转化为一维的问题。用 BFS 求出所有可达的点。如果过程中发现有环，而且环能到达终点，那么就是无穷个解。否则只要按照拓扑序进行 DP 就好了。由于这是一张无穷图，不可能全部 BFS，所以可以取 10^6 以内的（大概是 $4A^2$）。

时空复杂度	空间 $O(A^2)$，时间 $O(k2^k + A^2)$。

7.10

试题编号	JUNE12 MATCH ★
试题名称	Expected Maximum Matching

题目大意	算法讨论
一个 n,m 二分图，输入每条边存在的概率，求最大匹配数的期望值。 ($n \le 5, m \le 100$)	首先想到应用 Hall 定理。 考虑左边的状态，左边的点共有 2^n 个子集，分别记录这每个子集 X 的邻域集合 $f(X)$ 的大小是否不小于 X 的大小。每次新连边的时候，原来满足条件的子集依然满足条件，原来满足条件的子集并上一条新的边形成的集合也是满足条件的，这样就可以确定添边后的状态。 总共的状态数量其实是很少的，然后就可以用 dp，状态是 $f[i][j]$ 表示考虑右边的前 i 个点连过来的边，左边的所有子集状态是 j 的期望值。 由 Hall 定理，根据子集状态就可以直接确定最大匹配数了。

时空复杂度	空间 $O(m \times statenum)$，时间 $O(m \times statenum \times 2^n)$。

7.11

试题编号	MARCH12 CIELQUAK
试题名称	Ciel and Earthquake

题目大意	算法讨论
一个 $r \times c$ 的网格，相邻的格点之间有一条边。现在每条边以 p 的概率被断开，求出 $(1,1)$ 到 (r,c) 仍然连通的概率是多少。 ($r \le 8, c \le 10^{18}, 0.1 \le p \le 1$)	对于 c 较小的情况，可以采用经典的状压 DP。具体方法是按照轮廓线转移，状态是轮廓线上的点以及源点 $(0,0)$ 的连通性，用最小表示法来表示。转移的时候考虑新加进来的两条边是连着还是断开。与源点没有连通的状态可以舍弃。 对于 c 较大的情况，根据玄学可以知道答案 $f(r,c,p)/f(r,c-1,p)$ 的比值是收敛的，因此只要对 $c \le 50$ 暴力一下就可以了。

时空复杂度	空间 $O(\text{状态})$，时间：玄学。

7.12

试题编号	AUG12 MAGIC
试题名称	Two Magicians

题目大意	算法讨论
一个简单无向图上的顶点有两个人。轮流操作，每次操作包含三步骤：（1）沿着边任意走动或不走，如果能够走到对方的位置，则获胜；（2）添一条边但不允许重边。若无法添加，则对方胜；（3）放一次技能，可以瞬移到任一顶点。每人有 P 次技能使用机会。 $(t \leq 100, n \leq 7777, m, p \leq 10^5)$	考虑动态规划。记录当前状态为各自所在连通块的顶点个数奇偶性，剩下的技能次数，奇、偶大小的联通块各有几个，几条不产生连通性影响的边可以用。 只需要奇偶性而不需要具体数值，是因为这些不影响的边可以不停地轮流加直到剩余 ≤ 1 条。 然后通过某些手段可以得到这样的结论：（1）技能最多只使用 1 次。（2）在充分大的时候 DP 值关于奇偶连通块数量都是以 4 为周期循环的。 于是只要对小范围 DP 就可以了。

时空复杂度	空间 $O(n+m)$，时间 $O(n+m)$。

7.13

试题编号	OCT13 DEG3MAXT
试题名称	Three-Degree-Bounded Maximum Cost Subtree

题目大意	算法讨论
一个 n 个点，m 条边的带边权连通无向图，每个点双连通分量的点数不超过 9。求它一个子图，满足它是一个度数均不超过 3 的树，并最大化树的边权之和，并求出边权和最大时的方案数。 $(n \leq 100)$	首先求出所有点双连通分量，然后把图缩成若干个分量组成的树的形式。 对于每个分量，因为点数很小，所以可以状压 DP。状态中记录每个点目前的度数。转移时，为了保证不重复计数，可以强制每次剥去编号最小的叶子，这样一棵树只会有一种转移方式了。 然后对整个图缩成的树进行树形 dp 就可以了。方便的缩树方法是对于每个点双连通分量新建一个点，并将它连向分量中的所有点。在 DP 转移的时候要限制分量之间的割点的度数不要超过 3。

时空复杂度	空间 $O(2^{2k})$，时间 $O(n \times 2^{2k} \times k^2)$。这里 $k = 9$。

8 贪心

8.1

试题编号	MAY14 SEINC
试题名称	Sereja and Subsegment Increasings

题目大意	算法讨论
一个数组 $a[1..n]$，每次可以将连续一段全部增加 1，使得它最后在模 4 意义下与 $b[1..n]$ 相同，最少需要几次操作。 $(n \le 10^5)$	可以转化成初始值全部为 $a[i]=0$ 的情况。若没有模 4，那么容易证明答案为 $\sum \max(b[i]-b[i-1],0)$，所以现在就是要给某些 $b[i]$ 的值加上 4，使得最后的这个和尽量小。 考虑一段一段的加 4，如果给区间 $[l,r]$ 加 4，那么转化到差分数组 s 上就是 $s[l]$ 加 4，$s[r+1]$ 减 4。用贪心的思路，在差分数组上从左往右扫，当遇到 <2 的地方可以不用改，遇到 >=2 的可以考虑用前面的 +4 换这里的 -4 并使得总和尽量小。

时空复杂度	空间 $O(n)$，时间 $O(n)$。

8.2

试题编号	AUG11 SHORTCIR
试题名称	Shortest Circuit Evaluation

题目大意	算法讨论
给定一个布尔表达式，每个变量只出现一次。计算时满足短路原理。 现在知道每个变量取真值的概率，可以随意调换求值顺序，使得期望的计算次数最少。 $(len \le 30000)$	首先建出表达式树，这是经典问题，只要用两个栈就行。 表达式树的每个非叶结点可以有多个孩子，我们要做的是调整它们之间的顺序。 这可以用一个从底往上的树 DP 来做。对于每个结点 u，算出它取真值的概率 p，以及计算它所需要的最小期望次数 s。然后对所有孩子贪心排序，用简单的调整相邻两项的方法可以得知：如果是 or 的话就按照 p/s 排，如果是 and 的话就按照 $(1-p)/s$ 排序。

时空复杂度	空间 $O(n)$，时间 $O(n)$。

8.3

试题编号	NOV14 SEAORD
试题名称	Sereja and Order

题目大意	算法讨论
有 n 个程序，每个程序要在两台电脑上分别运行。第 i 个在第一台电脑跑 $a[i]$ 秒，在第二台跑 $b[i]$ 秒。一个程序同时只能在一台电脑上跑，一台电脑同时只能跑一个程序。求全部跑完的最小总时间是多少。构造方案。 $n \le 10000, a[i], b[i] \le 100000$。	容易猜测答案等于 $\max(\sum a[i], \sum b[i], \max(a[i]+b[i]))$。接下来用贪心（玄学）的方法构造：先按 $a[i]+b[i]$ 降序卡在两边，给中间留一条缝。然后按 $b[i]$ 升序塞在左边，但不重叠；再按 $a[i]$ 升序塞右边，也不重叠。剩下的分两种情况，按照 $a[i]-b[i]$ 降序或升序插在右边或左边。

时空复杂度	空间 $O(n)$，时间 $O(n \log n)$。

9 几何

9.1

试题编号	DEC14 DIVIDEN
试题名称	Divide or die

题目大意	算法讨论
给定一个角的顶点位置和两条边上的某点。保证角是 n 度，n 为整数。用尺规将这个角 n 等分。 $(0 < n < 360)$，操作数目不能超过 1000 步。	也就是说要作出 1 度角。 可以用奇怪的方法证明 $3\|n$ 时无解。 所以只要考虑 $\gcd(n,3)=1$ 的情况。我们可以用等边三角形作出 60 度角。可以用正五边形作出 72 度角。将它们的差 12 度角二等分再二等分，即可得到 3 度角。因为已知角不是 3 的倍数，所以可以用它和 3 的若干倍作差来得到 1 度角，问题就解决了。 等边三角形的作法很简单。 正五边形的作图关键是构造出 $\sqrt{5}$，这只要用直角边长分别为 1,2 的直角三角形的斜边即可得到。然后根据 $\sin 36^\circ = \frac{\sqrt{10-2\sqrt{5}}}{4}$，$1^2+(\frac{\sqrt{5}-1}{2})^2 = \frac{10-2\sqrt{5}}{4}$ 即可作出。

时空复杂度	空间 $O(1)$，时间 $O(1)$。

9.2

试题编号	SEPT13 TWOROADS
试题名称	Two Roads

题目大意	算法讨论
平面上有 n 个点，不存在三点共线。求出两条直线，每个点的代价是它到任一直线的最短距离的平方，使得总的代价最小。输出最小总代价。 $(n \leq 100)$	如果只有一条直线，这是经典的线性回归问题，可以通过求导数得到极小值，最后化出的式子中包含 $\sum x^2, \sum xy, \sum y^2, \sum x, \sum y, n$。 现在对于两条直线，以它交点处的两条角平分线为界，可以分成四块区域。对顶角的两块同属于一个。只要枚举区域的划分即可更新答案。本质不同的划分方法只有 $O(n^3)$ 个，只要固定一条直线贴紧的两点，再枚举与它垂直的那条线贴紧的点即可。让这条垂直线从远处扫过来，每跨过一个点可以 $O(1)$ 更新上面那些值。

时空复杂度	空间 $O(n)$，时间 $O(n^3)$。

9.3

试题编号	JULY13 RIVPILE
试题名称	Across the River

题目大意	算法讨论
一条河宽 w，河里面有 n 个桩子，给定坐标。有 m 种圆盘，每种无限多，且有一定的半径和单价。圆盘只能圆心安在桩子上。可以走通当且仅当圆盘有公共点。求过河的最小花费。($n, m \leq 250$)	对每个桩子拆成 m 个点，第 i 个点表示在这个桩子上安第 i 个圆盘，然后拆成出点和入点，中间连边费用为单价。 然后连桩子之间的边，暴力连是 m^2 条，需要优化。可以把每个连到对应的另一个桩子所需要的最小半径上，然后在每个桩内按半径升序连费用 0 的边，这样桩子之间的边就是 $O(m)$ 了。总共边数就是 $O(n^2m)$。然后用 dijkstra 跑最短路就行。

时空复杂度	空间 $O(n^2m)$，时间 $O(n^2m\log nm)$。

9.4

试题编号	JAN13 ANDOOR
试题名称	A New Door

题目大意	算法讨论
一个矩形区域内，给出 n 个圆，并将它们涂黑。求黑色区域的周长。($n \leq 1000$)	考虑每个圆的圆周上的弧对答案的贡献。 如果这个圆被其他圆包含了，那么不会有露出来的圆弧，贡献为 0。 否则，对于所有和它有交点的圆，相交部分覆盖的弧是没有贡献的，用余弦定理算出这段弧的两端对应的角位置。然后将所有端点的角位置排序，扫一遍，把没有被覆盖到的部分累加起来即可。 扫描一个环的时候要注意，越过起点的弧要将它拆成两个再加入。

时空复杂度	空间 $O(n)$，时间 $O(n^2\log n)$。

9.5

试题编号	OCT12 MAXCIR
试题名称	Max Circumference

题目大意	算法讨论
给出平面上一个三角形 ABC，和 n 个操作。第 i 个可以使 A 沿 (x_i, y_i) 平移。 最多使用 k 次操作，每种只能用一次，最大化三角形 ABC 的周长。 ($n \leq 500, \lvert x_i\rvert, \lvert y_i\rvert \leq 10^6$)	BC 边长度不变，所以只要考虑 $AB + AC$。 如果答案为 l，那么 A 点在 B, C 为焦点，长轴为 l 的椭圆上。令椭圆上 A 点处的法向量为 p，那么选中的操作向量 s_i 一定使得 $(\sum s_i) \cdot p$ 最大，也就是取了 $s_i \cdot p$ 最大且为正值的至多 k 个向量。 那么枚举 p 的方向，同时维护所有向量的投影的大小顺序。p 绕一圈的过程中会有 $O(n^2)$ 次顺序交换，发生在 p 与 $s_i - s_j$ 垂直时。把这些时间点排序再扫即可。 由于本题精度要求很高，求平方根时需要把整数和小数部分分开算，才能保证绝对精度。

时空复杂度	空间 $O(n^2)$，时间 $O(n^2\log n)$。

10 玄学和暴力

10.1

试题编号	APRIL15 LPARTY
试题名称	Little Party

题目大意	算法讨论
给定一个由 M 个元素组成的集合，每个元素由 N 个布尔变量组成，每个变量可以为真或假。要求找出一个总长度最小的基集合，使得以这个基集合中元素为子集的元素的集合恰好为给出的集合。基集合中元素由 N 个变量中某些变量组成，每个变量也仍然可以为真或假。给出的 M 个元素一定包含所有 N 个变量。 $(t \leq 120, n \leq 5)$	首先有 3^5 个备选集合。但是其中有大量是不合法的，先把它们删掉。 然后其中有一些集合是互相包含的，在两者间仅保留小的。 然后还有一些集合是仅相差一位的，那么这两个集合都可以删去，因为他们去掉那个相差一位后得到的集合仍然满足条件。 那么剩下来的备选集合就很少了，可以直接爆搜。 搜的时候需要注意可行性剪枝、最优性剪枝，以及适当的搜索顺序。

时空复杂度	空间 $O(2^{2^N})$，时间玄学。

10.2

试题编号	JAN15 RANKA
试题名称	Ranka

题目大意	算法讨论
在一个 9×9 的围棋棋盘上下棋，可以 pass，不能出现重复的棋盘状态。构造一个 10000 步的方案。	9×9 的棋盘上可以摆 8 个劫，还剩下几十个无关的空格子。用这个 8 个劫可以构造 2^8 个状态。由于每次只能将其中一位异或上 1，因此可以用格雷码的方法构造。再乘上剩下的几十个格子所形成的状态，就可以达到 10000 步了。

时空复杂度	无

10.3

试题编号	APRIL14 GERALD08
试题名称	Chef and Tree Game

题目大意	算法讨论
一个有根树，每条边是蓝或红色。两人轮流操作，先手可以删红边，后手可以删蓝边。删边之后与根不连通的部分也被删去。不能操作者输。判断胜负情况。 $(n \leq 100000)$	这是 Blue-Red Hackenbush 问题，构造 surreal number 来解决。然后就是一个树 DP，从底向上计算函数值。 问题中涉及的函数值是分母为 2 的幂次的有理数。计算过程中需要实现右移和加法操作。由于位数较多，double 不够用。可以用一个 set 存下二进位为 1 的数位的下标。 右移时可以整体打偏移量标记。 加法操作时会有进位。可以用启发式合并，将位数少的数中的 1 一个个加到大的上面，如果对应位置已经填有 1，则不断往前进位即可。

时空复杂度	空间 $O(n \log n)$，时间 $O(n \log^2 n)$。

10.4

试题编号	JUNE13 TKCONVEX
试题名称	Two k-Convex Polygons

题目大意	算法讨论
从 n 个棍子中选出 $2k$ 个，拼成两个凸 k 边形，是否存在方案。 $(n \le 1000, k \le 10)$	首先，能拼出的充要条件是最长边的长度小于剩下所有边的和。 先将棍子按长度排序，然后枚举两个多边形的最长边。从剩下的边中挑选尽量大的边。假设下标分别是 $i < j$，当 $i \le j - k$ 时，方案显然是各自取比它小的最大那 $k - 1$ 条。如果 $i > j - k$，那么中间就有重叠的部分，需要决定哪些边分给哪个多边形，这只要暴力枚举就行了。

时空复杂度	空间 $O(n)$，时间 $O(n^2 + n\binom{2k}{k})$。

10.5

试题编号	FEB13 ROC
试题名称	Room Corner

题目大意	算法讨论
房间是一个边界水平或竖直的多边形。每个 90 度内角处的内部格子都站着一个小朋友。每次相邻的两个小朋友可以交换位置，速度都是每秒一格。每次给出一对小朋友，询问他们两个发生相遇的最短时间。 询问次数 ≤ 10000。地图大小 $\le 2500 \times 2500$。每行里的房间内部格子只能形成一个连续段。	所有小朋友最后会连成一个环。显然这个环是沿着房间的边界的。我们任取一个小朋友，让他沿着墙逆时针绕着房间走一整圈，那么就相当于把环给遍历了一遍。 当 A 和 D 想要相遇，而相遇发生在相邻两点 B, C 之间（排列顺序为 A, B, C, D），那么所需的最小时间为 $\max(AB, CD) + BC/2$。注意到如果设 BC 中点为 M，上式即等价于 $\max(AM, MD)$。因此 M 越接近 AD 的中点 N 越好。 所以我们可以把环上每条边的中点按顺序存在数组里，对于每个询问 A, D，用它们的中点 N 去数组里二分，寻找最近的点即可。

时空复杂度	处理整个地图的时间和空间是 $O(n^2)$ 的，n 是地图边长。 回答每次询问是 $O(\log m)$ 的，m 是小朋友的个数。 注意由于题目限制，每行最多只有两个角落格子，因此 $m = O(n)$。

10.6

试题编号	MARCH12 EVILBOOK
试题名称	Evil Book

题目大意	算法讨论
有 n 个厨师，对于第 i 个可以用 c_i 点血干掉他然后得到 m_i 点魔。用 x 点魔可以将某个厨师的 c_i 和 m_i 都乘以 1/3。要收集齐 666 点魔至少需要花费几点血。 $(n \le 10, 10 \le x < 666, 0 \le c_i, m_i, \le 10^9)$	这是一道暴搜题。 每次选择一个还没有被干掉的厨师，枚举对它使用几次技能。注意使用技能的次数应该保证花掉的魔比得到的魔要少，否则肯定不是最优的。另外，如果在得到的魔 > 666 的情况下，显然能完成任务，这时要使得花费的血尽量少。 根据上面这些剪枝，每层的分支数目是不多的，可以较快搜出来。

时空复杂度	空间 $O(n)$，时间：玄学。

10.7

试题编号	JUNE12 COOLNUM
试题名称	Cool Numbers

题目大意	算法讨论
一个数有 k 位，每个数位上的数字分别为 $X_1, X_2, \cdots, X_K$，如果一个数 n 存在一到三个数位上的数的和为 s，且 $(X_1 + X_2 + \cdots + X_K – S)^S$ 是 n 的倍数，那么 n 是 cool number。定义 $LC(N)$ 和 $RC(N)$ 分别是小于等于 N 的最大的 cool number 和大于 N 的最小的 cool number，多次询问给定 N，求 $LC(N)$ 和 $RC(N)$。 $(1 \leq t \leq 10^5, 1 \leq n \leq 10^{1000}, \sum len \leq 4 \times 10^6)$	首先如果一个数只有不超过 3 个非零位，显然是可以的。这是 trivial 的情况。用 $O(len)$ 的时间可以找出它左右的 trivial 答案，只要保留前 3 个非 0 位，然后搞一搞就行了（求右边的话要给最后一个非 0 位增加 1），当然如果只有 2 个或以下，那就将个位加 1 就可以了。 nontrivial 的数字很少，可以爆搜。搜的时候按照数位和去搜，然后遍历它的 27 次方的所有约数，然后判断。因为数越大越难满足，数量就越稀疏，所以把较大的数本地搜好打表即可。 然后询问的时候在列表里二分即可。

时空复杂度	空间玄学，时间玄学。

10.8

试题编号	APRIL12 CONNECT
试题名称	Find a special connected block

题目大意	算法讨论
一个 $n \times m$ 的方格图，每个格子或者是障碍，或者有非负整数。格子有正的费用。选出一个四连通块，使得块中至少包含 k 种不同的正整数，且费用之和最小。 $(n, m \leq 15, k \leq 7)$	如果整数标号的范围很小，问题会方便一些。所以考虑把所有整数随机映射到 $[0, k-1]$ 的整数上然后再做。现在变成要求连通块包含 $[0, k-1]$ 的所有数字。这是一个经典的斯坦纳树问题：对包含的数字压位，有两种转移方式，一种是添子树，一种是连父亲。前一种用枚举子集更新，后一种用最短路。 这样得到的并不一定是最优解。但多随机染色几次，就容易碰到适合最优解的染色方式。

时空复杂度	空间 $O(nm)$，时间 $O(times \cdot (nm3^k + 2^k nm \log nm))$。

10.9

试题编号	NOV11 DOMNOCUT
试题名称	Colored Domino Tilings and Cuts

题目大意	算法讨论
给出一个 $n \times m$ 的矩形棋盘。构造一个 1×2 或 2×1 骨牌的不重叠不遗漏的覆盖。使得棋盘的割尽量少。在此基础上骨牌的色数尽量少。 割是垂直或水平且不穿过任何骨牌的直线。 色数是使有公共边的两个骨牌的颜色不同所需要的最少颜色数量。 $(n, m \leq 500)$	首先格子数量需要是偶数。 然后对于 $n \leq 4$ 的情况，都可以轻易的给出构造。 对于 n, m 较大的情况，首先需要发现两个基本棋盘，一个是 5×6 的，另一个是 6×8 的，它们都不存在割，而且只需要 3 种颜色即可染色。这两个棋盘可以通过暴搜或者人类智慧得到。 对于 n, m 一奇一偶或者全为偶数这两种情况，都可以由上面的基本棋盘通过扩展而得到。扩展一次可以使行数或列数增加 2，之后得到的棋盘依然是不存在割，且色数为 3 的。方法是每行（或列）插入一个横放（或竖放）的新骨牌。

时空复杂度	空间 $O(nm)$，时间 $O(nm)$。

10.10

试题编号	OCT11 BAKE
试题名称	The Baking Business

题目大意	算法讨论

一个多维数组的插入和求和。
插入格式为
 	
`I product_id[.size_id] province[.city_id[.region_id]] M/F age units_sold`
 	
询问格式为
 	
`Q product_id[.size_id] province[.city_id[.region_id]] M/F start_age[-end_age]`
 	
具体细节可以参考原题面。
(操作数目 $\leq 10^5$)
	这是一道阅读理解题和读入处理题。
	直接开一个七维数组模拟操作即可。对于每一维需要多开一位表示这一维不带限制时候的和。然后由于有子段查询，需要按年龄维护前缀和。

时空复杂度	空间 $O(size)$，时间 $O(n \times (100 + 4 \times 3))$。

11 Challenge 题

11.1

试题编号	NOV14 SPELL
试题名称	The Spelling Problem

题目大意	算法讨论
给你一个字典和一个英语文章。 英语文章中的单词可能出现错误。错误的方式有：交换两个字符，漏掉一个字符，多出一个字符，敲错一个字符。 你的任务是找出文章中的错误并改正，并尽量最大化你的得分。分数是这样算的：把原来错的单词改对可以加 3 分，对的改错会扣 1 分。 文本长度不超过 10MB。	题目中的这个字典会很大，包含太多的生僻词。一个较好的解决办法是找份常用词列表放到程序里（比如牛津 3000）。改词的时候尽量往常用词改，不要改成生僻词。 然后就遍历这篇文章的每个单词，首先判断这个词是不是正确词，给它一个估价，如果估价足够大就认为正确。常用词的估价可以设的大一些，其他的非常用词就设的小一些。另外注意在文章语境中频繁出现的词语，也要给它更大的估价。 如果是正确词就输出来。否则就枚举每一种修改方法，并从中选出估价最大的那种并进行修改，然后更新这个词的估价。 枚举修改方法可以用 hash 函数和 trie 树配合实现。

时空复杂度	空间 $O(nl)$，时间 $O(nl^2)$。

11.2

试题编号	FEB13 EFFPAINT
试题名称	Efficient Painting

题目大意	算法讨论
你有一块 $n \times n$ 的正方形的画布，初始时每个格子都是白色的。你每次可以选择一个边必须平行于坐标轴的子矩形并对它进行操作。 White - 矩形内全涂成白色。 Black - 矩形内全涂成黑色。 Flip - 矩形内的白色变成黑色，黑色变成白色。 你会拿到一个所要求的最终图案。你需要用尽量少的操作次数，将画布上的图案变成所要求的样子。用的次数越少则获得的分数越多。 数据生成与评分方式如下：$10 \le n \le 50$ 共有 50 个数据文件，每个都是这样生成的：一个整数 n 从 $[10, 50]$ 中均匀随机抽取。一个实数 p 从 $[0.4, 0.6]$ 中均匀随机抽取。然后每个格子都是独立的以 p 的概率填为黑色，以 $1-p$ 的概率为白色。	考虑所有 $(n+1)^2$ 个格点，一个格点周围有 4 个格子（边界的只有两个，四角的只有一个）。给这个格点赋一个权值，为周围格子中黑色的数量的奇偶性。 一个矩形进行 F 操作对格点的权值所产生的影响：对于矩形内部和边界上的格点，黑格数量奇偶性不变；对于四角上的格点，奇偶性也会反转。也就是说，一次 F 操作可以将矩形四个顶点的权值反转。 计算出终盘上各个格点的权值，考虑倒过来操作将其中的 1 全部变为 0，得到全空白的画布。那么一次 F 操作至多可以消去 4 个 1；如果无法同时消去 4 个 1，就消去 3 个 1，添回一个 1。 于是我们要寻找盘面上四个顶点都是 1 的矩形。这可以做到每次查找 $O(n^3)$：对第 i 行，枚举 $j < k$，使得 $w[i][j] = w[i][k] = 1$，并将 (j, k) 标记为可行；如果对另外一行也有 (j, k) 可行，就找到了这样的矩形。 如果找不到，就找顶点有三个 1 和一个 0 的矩形，方法也和上面类似。 容易证明这个贪心方法一定能求得一个合法解。

时空复杂度	空间 $O(n^2)$，时间 $O(n^5)$。

11.3

试题编号	AUG13 DELNMS
试题名称	Deleting numbers

题目大意	算法讨论
假设当前有 n 个数形成的数组 $a[n]$，每次可以选择 v,t 两个数，满足 $v+t \le n+1$，且令 k 为最大的满足 $v+kt \le n$ 的数，必须满足 $a[v]=a[v+t]=a[v+2t]=...=a[v+kt]$。然后这些数将会删除 (若 $v+t=n+1$ 则只删除第 v 个数)，之后形成新的由 $n-k$ 个数形成的数组 $a[n-k]$，满足这个数组的元素是原来数组被删除之后剩下的元素，且相对位置保持不变。 设计一种方案删除所有的数，你的得分与你的方案的总步数有关。	这个题目的操作并不是特别好处理，因为删去一个等差数列后剩下的数字会被重新排列回去，之间的相对位置会发生改变。 一种比较简单好写的做法是倒序删除，这样需要 n 步。 对于小规模的数据，可以采取限定深度然后暴搜的方法。每次枚举一个等差数列删掉，看看怎样效果比较好。 对于大规模的数据，可以考虑先把出现一次的数字删完，然后剩下的都是至少出现多次的数了，可以从后往前删，删的时候如果能往前扩展成等差数列的话就更好。或者直接保留出现最多次数的数字，一次性删掉。 一个优化是从后往前分成好多段，对每一段内进行上述做法。

时空复杂度	空间 $O(n)$，时间 $O(n)$。

11.4

试题编号	JUNE12 CLOSEST
试题名称	Closest Points

题目大意	算法讨论
三维空间中有 n 个点。 共有 q 个询问，每次给定一个点，询问 n 个点中离这个点最近的是哪个，你需要输出它的编号。 $(n,q \le 50000)$	这是一个经典问题。经典的做法是使用 kd 树来优化查询。 这题中需要注意建树时需要按照方差最大的那一维进行划分。另外还要注意算方差和算距离时要开 double 或者 unsigned long long，不然会爆。

时空复杂度	空间 $O(n)$，时间 $O(qn^{2/3})$。

11.5

试题编号	JUNE13 CHAORNOT
试题名称	To challenge or not

题目大意	算法讨论
请你找这样一组整数，里面没有 3 个数构成等差数列。 让问题更难些，你必须从给定的 M 个整数 $B_1, B_2, \cdots, B_m$ 中选取。 不一定要最大数集，但是数集越大分数越高。 $(n \le 100000)$	这个题最简单的想法是直接贪心。随机一个顺序，然后往里面加，如果能加入就加，不能加入就跳过。 怎么判断能不能加入呢？开一个数组 $b[]$，把当前不能加入的数字 x 设为 $b[x]=1$。 每次加入一个新数字时，需要更新 $b[]$ 数组，只要枚举之前加过的数字，把它们的等差中项（或第三项）设为 $b[x]=1$ 即可。 这个策略每次搞一遍是 $n+len^2$ 的。事实上这个策略可以拿到较高的分数。 考虑取得更高的分数的话，可以多次随机顺序然后取最优解。

时空复杂度	空间 $O(n)$，时间 $O(k(n+len^2))$。

11.6

试题编号	DEC14 KALKI
试题名称	Kali and Devtas

题目大意	算法讨论
给定二维欧氏平面内的 N 个点，你需要返回这些点的一个生成树，使得 C_i 的最大值最小。 C_i 的定义是：对于每个点，设在生成树中与其相邻的点中最远的点的距离为 R，那么以该点为圆心，半径 R 以内的的点的 C_i 全部增加 1（包括自身）。 $(n \le 400)$	这个题是很难的一个题。最好的办法是写一些跑的过时限的 simple 做法，然后取个较优的答案。 首先可以考虑贪心，建树的时候考虑加入某个新点，然后这个点和树上已有的某个点尽量靠近，这样的解看起来会比较优秀。 然后进一步思考，可以发现上面这个想法和最小生成树比较类似。于是可以写一个平面完全图的最小生成树。 这个东西可以用 V 图来做，但是由于数据范围很小，直接暴力也不虚。

时空复杂度	空间 $O(n)$，时间 $O(n^2 \log n)$。V 图求最小生成树的话是 $O(n \log n)$。

11.7

试题编号	OCT12 MAXRECT
试题名称	Maximum Sub-rectangle in Matrix

题目大意	算法讨论
给出一个 $H \times W$ 的整数矩阵 A，行标号 $0 \sim H-1$，列标号 $0 \sim W-1$，求一个子矩阵使其中元素之和尽可能大。 这个子矩阵不要求是连续的。 $(H, W \le 300)$	首先考虑，如果选择的行集合是固定的，那么很容易求出此时选取某一列时得到的收益，那么只要把所有收益为正的列给取出来就好了。 选择行集合也要自己选，那么我们就得到了一个 $2^H \times W$ 的做法。 然而这题中 H 比较大。我们可以采取模拟退火方法，随机地选出一行把它加到当前集合里，如果答案增加了就保留，否则按一定概率舍弃。 这样已经能得到很高的得分了。

时空复杂度	空间 $O(HW)$，时间 $O(HW + W \times t)$，其中 t 是退火次数。

11.8

试题编号	SEP12 SIMNIM
试题名称	Simultaneous Nim

题目大意	算法讨论
有 n 个异或和为 0 的整数 a_i。 你需要把它们分为 k 个集合，使得每个集合内的数字的异或和也为 0。 k 越多越好。 $(10 \le n \le 1000, a_i \le 2^{60} - 1)$	首先因为这些数的异或和为 0，所以如果从中抽出一个集合的异或和为 0，那么它的补集的异或和也是为 0 的。于是只要考虑不停地从原集合中分出一个异或和为 0 的集合就好。 为了让分出的集合数目尽可能多，直观来说应该让每次分出来的那个集合尽可能小。 如果能挑出一个线性相关的真子集，那么就可以从中找出一个为异或和为 0 的集合。所以可以从当前集合中随机挑出一个大小为 m 的子集，然后消元判一下是否有解，如果无解就接着随机，或者把 m 放大，直到找出一个集合为止。这样不断做下去就可以得到比较好的得分。

时空复杂度	空间 $O(n)$，时间 $O(nm \times t)$，其中 t 是随机次数。

11.9

试题编号	NOV11 STEPAVG
试题名称	Stepping Average

题目大意	算法讨论
考虑以下的方法求出 N 个数的“迭代平均数”。每次拿出其中任意两个数，并用他们的平均数取代他们，重复 $N-1$ 次直到最后只剩一个数。我们叫这个剩下的数叫“迭代平均数”。值得注意的是，不同的合并顺序最后会导出不同的“迭代平均数”。 给定 N 个数。找到一个合并的顺序使得“迭代平均数”尽量接近给定的数 K。 $(N \leq 1000)$	考虑一个最 naive 的合并方法，从后往前不断合并，比如 a, b, c, d 先合并 c, d，再和 b 合并，最后和 a 合并。得到的答案为 $a/2 + b/4 + c/8 + d/8$。 如果 N 越来越大，那么后面的数字所占的权重就指数级变小。所以我们只要考虑前面的几个数字的贡献。 最简单的是采取随机的方法，从 1/2 的往后逐个确定。确定某一位上的数字时，尝试将后面其他数与它调换，看看答案会不会更接近 K。由于权重是以 1/2 的比例不停减小的，所以这样子的策略可以使得与答案相差 $0.000\cdots1$ 以内。 这样的策略就可以在 codechef 上拿到满分了。

时空复杂度	空间 $O(n)$，时间 $O(nkt)$，其中 t 是随机次数，k 是只调整前 k 个位置。

11.10

试题编号	APRIL12 SIMGRAPH
试题名称	Similar Graphs

题目大意	算法讨论
给你两个点数都为 n 的无向图。你需要给图的顶点重新标号，使得它们的公共边数尽可能多。公共边定义为二元组 (a, b) 使得两个图都满足其中标号为 a 的点和标号为 b 的点有连边。 $(n \leq 75)$	这是一个非常困难的问题。 首先把第一张图的标号固定为 $1 \sim n$，那么只要考虑第二张图的标号就好了。 最 naive 的做法是随机若干发，然后取一个答案最大的排列。但这样的解太差。可以考虑在这个解的基础上进行模拟退火。每次随机选出两个点，并在第二张图中把这两个点编号交换，然后可以用 $O(n)$ 的时间求出答案的变化。如果答案变优了就保留，否则就按一定概率舍弃这个解。 这样就可以得到较高的得分了。

时空复杂度	空间 $O(n^2)$，时间 $O(n^2 + tn)$，其中 t 是随机次数。