2016国家集训队作业试题泛做

安徽师范大学附属中学 罗哲正

2015 年 12 月 25 日

一共包含100道试题(90道传统+10道Challenge)的泛做表格，试题按照时间顺序排列。

Problem 1

试题名称	Attack of the Clones		
试题来源	Codechef JUNE 11	试题编号	CLONES
题目大意		算法讨论	
称一个映射$f: A \to B$为布尔函数要求如下：A是长度为n的01串组成的集合，$B = \{0, 1\}$，n为布尔集合的项数。 我们定义以下四种n项布尔函数集合： 1.Z：所有满足$f(0, 0, \cdots, 0) = 0$的集合。 2.P：所有满足$f(1, 1, \cdots, 1) = 1$的集合。 3.D：满足$!f(x_1, x_2, \cdots, x_n) = f(!x_1, !x_2, ..., !x_n)$的集合。 4.$A$：满足如下条件：若$f(x_1, \cdots, x_{i-1}, a, x_{i+1}, \cdots, x_n) = f(x_1, \cdots, x_{i-1}, b, x_{i+1}, \cdots, x_n)$ 则$f(y_1, \cdots, y_{i-1}, a, y_{i+1}, \cdots, y_n) = f(y_1, \cdots, y_{i-1}, b, y_{i+1}, \cdots, y_n)$ 的集合。现在给出一个由$Z, P, D, A, \vee, \wedge, !, /, (,)$组成的集合表达式$S$。其中$\vee, \wedge, !, /$ 分别表示并集，交集，补集和差集。优先级为$() >! > \vee, \wedge, /$。求这个表达式结果集合的元素个数。 T为数据组数，$T \le 100, \|S\| \le 100$.		我们考虑一个函数和四个集合Z, P, D, A的从属情况，可以把集合分成16种，我们手算出满足某些条件16中情况的集合方案数，然后子集反演出恰好满足这些条件的集合方案数。而这些部分是否会属于表达式的结果只有2^{16}情况，于是就可以用一个16位二进制数来表示表达式运算的值。 于是对于两个集合X, Y，并集，交集，补集，差集就分别变成了$X\,or\,Y, X\,and\,Y, X\,xor\,65535, X\,xor\,(X\,and\,Y)$，剩下的就是简单的表达式处理了。	
时间复杂度	$O(T\|S\|)$	空间复杂度	$O(\|S\|)$

Problem 2

<table>
<tr><td>试题名称</td><td colspan="3">Minesweeper Reversed</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 11</td><td>试题编号</td><td>MINESREV</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定你一个$R*C$ 的扫雷棋盘，其中的雷的位置已经表明。最开始所有的方块都是打开的，你需要关闭所有的方块。你可以通过一次点击来关闭一个方块(可以关闭含雷的方块)。在你关闭(x, y)后，在正常的扫雷游戏中可能和(x, y) 同时被打开的格子都会被关闭。现在要你求出至少点击多少次，可以关闭所有的方块。
T是数据组数，$T \le 50, 1 \le R, C \le 50$.</td><td colspan="2">每个雷都需要一次点击。我们把剩下来的格子分成两类，第一类是和雷相邻的，第二类类是不相邻的。第二类的格子组成了若干个联通块(注意这里指的是八连通)，由于每个第一类格子最多和两个第二类联通块相邻。显然不与任何第二类联通块相邻的第一类格子必须单独一次点击来关闭。
接下来我们可以把第二类联通块当成顶点，把连接两个不同的第二类联通块的第一类格子当成边。建出一个无向图。每次可以选择点击边或者点，点击边就一次关闭两个联通快，所以我们求出这张图的最大匹配，再用点数减去最大匹配数就是关闭剩余的格子的最小点击次数。
一般图最大匹配是经典问题可以使用带花树解决，由于这是一个平面图，点数和边数都是$O(RC)$的，且常数很小。</td></tr>
<tr><td>时间复杂度</td><td>$O(TR^2C^2)$</td><td>空间复杂度</td><td>$O(RC)$</td></tr>
</table>

Problem 3

<table>
<tr><td>试题名称</td><td colspan="3">Billboards</td></tr>
<tr><td>试题来源</td><td>Codechef JULY 11</td><td>试题编号</td><td>BB</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有一个长度为n的01序列，它任意长度为m的连续子序列中都有k个1。问所有满足条件的01序列中，1的数目最少的不同的序列有多少个？
答案对10^9+7取模。
$1 \le k \le m \le 50; m \le n \le 10^9$.</td><td colspan="2">首先考虑m整除n的情况，我们把串分成$\frac{n}{m}$段，每一段长度都是m，那么每一段中至少有k个1，整个串至少有$\frac{kn}{m}$个1，每一段后k个放1是满足条件的，所以1的个数至少是$\frac{nk}{m}$。现在对每个序列构造一个$k * \frac{n}{m}$的矩阵A，矩阵的第i行第j列表示在第j段中第i个1在这一段中的位置。如果一个序列合法，那么每一行是单调不增的，每一列是单调增的。显然这是一一对应的。
这是一个版标准杨氏矩阵，方案数$ans = \prod_{i,j} r+j-ihook(i,j)$其中$r$表示候选集合大小，$hook(i,j)$表示和$(i,j$同行或者同列至少有一维坐标比它大的位置合数。不过这个矩阵太大，我们把分子和分母列出来的话会发现大多数元素都可以约分掉，只剩下$O(km)$个数需要暴力算。
当m不整除n的时候，如果$n \mod m \le m-k$，则每一组的前$n \mod m$都是0，否则每一组的后$m-(n \mod m)$都是1把确定的部分去掉之后有转化成了$m|n$的情况，同上计算即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(mk)$</td><td>空间复杂度</td><td>$O(1)$</td></tr>
</table>

Problem 4

试题名称	Trial of Doom		
试题来源	Codechef JULY 11	试题编号	YALOP
题目大意		算法讨论	
给定一个$n*m$ 的网格，每一个格子为蓝色或者红色，一共有k 个红色格子，你可以在这个网格里走，每当你离开一个格子的时候，这个格子和它的四周四个格子会改变颜色，现在让你判断是否存在一条左下角到右上角的可以经过相同格子的路径，使得从左下角开始，沿着这条路径走，走到右上角再离开这个网格后，网格里所有格子都变成了蓝色。 T为数据组数，$T \le 50; n,m \le 10^9; \min(n,m) \le 40; k \le 10000.$		不妨设$n \le m$当$n = 1$的时候路径长度的奇偶性和n一定相同所以可以枚举第一个格子经过次数的奇偶性，然后解出每个格子经过多少次，再判断一下是否是合法路径。 当$n > 1$的时候，对于任意格子集合S都存在一条路径经过S集合内每个元素奇数次，其他格子偶数次，所以我们只要列出方程判断是否有解即可。 考虑把所有的红格子移动到一行，格子(i,j)是红格子等价于$(i-1,j-1),(i-1,j),(i-2,j),(i-1,j+1)$是红格子，这样就能把第$j$列移动到第$j-1$列上。 因为$n$很小，可以发现存在一个周期$C$是的第$i$列的格子对第$j,j-1$列的格子的影响与对第$j-C,j-C-1$列的影响是一样的，我们可以暴力预处理这个循环节，就可以在$O(nk)$时间内把所有的红格子放到第一列，这样就得到了n个方程，判断是否有解即可。一列格子可以使用long long压位存储。	
时间复杂度	$O(Tk+TnC)$	空间复杂度	$O(k+C)$

Problem 5

<table>
<tr><td>试题名称</td><td colspan="3">Something About Divisors</td></tr>
<tr><td>试题来源</td><td>Codechef AUG 11</td><td>试题编号</td><td>DIVISORS</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">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$D(N < D \le B)$ 能整除NX。
T为数据组数，$T \le 40; X \le 60; B \le 10^{12}$.</td><td colspan="2">令正整数$i = \frac{NX}{D}$，那么显然有$i < X$，枚举每一个可能的I，为了避免重复，我们可以计算出满足$i|NX$且不存在j满足$i < j < X$且$j|NX$的N。
$i|NX$，于是$\frac{i}{\mathrm{gcd}(i,X)}|N$，令$A_i = \frac{i}{gcd(i,X)}$，就有$N = A_i p$。因为$N \le \frac{B_i}{X}$所以$p \le \frac{B_i}{XA_i}$，设$P$为这个取值的上界。
因 为$j|A_i X p$，有$\frac{j}{\mathrm{gcd}(A_i,X,j)}|M$，令$B_j = \frac{j}{\mathrm{gcd}(A_i,X,j)}$。则满足$i|NX$的同时满足$j|NX$的条件是$D_i|M$。我们可以枚举$i$，然后对$j$进行容斥，即$ans_i = \sum(-1)^t \frac{P}{lcmB_j}$。直接计算会超时，但是当分子超过$P$时答案就是0，而且可能的lcm值并不多，于是就可以用DP来计算，注意常数优化。</td></tr>
<tr><td>时间复杂度</td><td>$O(X^2K)$</td><td>空间复杂度</td><td>$O(XK)$</td></tr>
</table>

Problem 6

<table>
<tr><td>试题名称</td><td colspan="3">Shortest Circuit Evaluation</td></tr>
<tr><td>试题来源</td><td>Codechef AUG 11</td><td>试题编号</td><td>SHORTCIR</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">布尔表达式满足短路运算原理，即如果某个表达式已经得到正确的结果，那么就不再继续计算这个表达式。现在给出这个表达式S以及n个变量(只出现一次)为真的概率p_i，你可以在表达式本质不变的情况下调整这个布尔表达式的顺序，使得期望的计算次数最少，求最小期望次数。
$|S| \le 30000; n \le 1000; 0 < p_i < 1$.</td><td colspan="2">我们先建立一棵表达式树，其中每个节点的所有孩子都用同一种运算符连接。所以只要调整每个节点的孩子顺序即可。我们对于每个节点维护两个量f_i表示期望运算次数，e_i表示为真的概率。显然，当or连接的时候按照$\frac{f_i}{e_i}$升序排列，当and连接的时候按照$\frac{f_i}{1-e_i}$升序排列最优。所以做一个树形DP即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(|S|\log|S|)$</td><td>空间复杂度</td><td>$O(|S|)$</td></tr>
</table>

Problem 7

试题名称	Counting Hexagons		
试题来源	Codechef SEPT 11	试题编号	CNTHEX
题目大意		算法讨论	
你有NK根木棍木棍长度为1到N且每种长度的木棍有K根。你需要选出六根木棍拼出一个面积为正的六边形。你选取的木棍需要满足：最长的木棍长度至少为L，其它的木棍长度不能超过X。求方案数对10^9+7取模，两个方案是不同的当且仅当存在一个长度的木棍在两个方案中的选取个数目不同。 $1 \le K \le 5; 1 \le X < L \le N \le 10^9; N-L \le 100$.		$N-L$很小所以可以直接枚举最长的木棍，要求其他木棍的长度和大于最长木棍的长度。 我们通过一个数位DP求答案。设$f_{a,b,c,d,e}$表示考虑了后a位，当前其他木棍长度和与最长的木棍大小关系为b，进位为c，这五根木棍的大小情况为d，这些木棍的后a位与X 的大小关系为e。在二进制下DP的话只要枚举第$a+1$位是0还是1即可。 为了避免出现长度为0的木棍，可以先把$X-1$最长木棍长度-1，最后每个木棍+1。 令$\alpha = max(K) = 5$。	
时间复杂度	$O(4\alpha^2(N-L)\log N Bell_\alpha)$	空间复杂度	$O(\alpha(N-L)2^\alpha Bell_\alpha)$

Problem 8

试题名称	Short		
试题来源	Codechef SEPT 11	试题编号	SHORT
题目大意		算法讨论	
给定n,k求满足$n < a,b < k$的数对(a,b)数目使得$(a-n)(b-n) \mid (ab-n)$。 $0 \le n \le 10^5; n < k \le 10^{18}$.		设$a > b$可以得到$b = n + \frac{n(a-n)}{p(a-n)-a}$，我们可以枚举所有可能的$a$再枚举$n(a-n)$的所有约数$d$，可以求出$p = \frac{d+a}{a-n}$和$b$，判断合法就加入答案。 当$a \le b$时，可以发现$d + a > 2(a-n)$，所以$b \le n + \frac{n(a-1)}{a-2n}$，而$a \le b$，所以$a \le 2n + \sqrt{2n^2-n}$。枚举$a$即可。至于枚举$d$，我们可以预处理所有可能的$a-n$进行质因数分解，然后直接枚举$n(a-n)$的约数来判断，但是还是不能在时限内完成。 考虑当$a$比较大的时候，因为$a \le b$可以得到$p \le \frac{a^2-n}{(a-n)^2}$，所以可以枚举所有可能的$p$，解出$d$判断合法即可。 设$U = 1177$为切换枚举方法的阈值。	
时间复杂度	$O(TnU)$	空间复杂度	$O(n\log n)$

Problem 9

<table>
<tr><td>试题名称</td><td colspan="3">The Baking Business</td></tr>
<tr><td>试题来源</td><td>Codechef OCT 11</td><td>试题编号</td><td>BAKE</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有n个操作，每一个操作都可能是增加订单或者询问，每一个订单用如下的方式描述：
I 产品编号[大小编号] 省编号[城市编号[地区编号]] 性别 年龄 出售数
顾客的性别为M 或者F，年龄从1到90。注意所有的编号都是从0开始。还需要注意方括号内的部分是可选的，因为某些出售的这些信息丢失了。出售数量不会超过100。每一个询问都可以用如下的方式描述：
Q 产品编号[.大小编号] 省编号[.城市编号[.地区编号]] 性别 起始年龄 [-结束年龄]
询问该限制下的出售总数。如果方括号内的信息缺失说明对这个信息没有限制，特殊地如果产品编号省编号是-1说明没有限制。
有10种产品，每种都有3种不同的大小。有10个省份，每个省份可以被划分为20个城市，每个城市又可以被划分成5个地区。$n \le 10^5$</td><td colspan="2">直接开一个七维数组来维护，增设一个单元表示该元素缺省时的计数。插入元素时人工讨论输入信息删去所有可缺省信息的方案。询问时也是人工讨论，由于需要支持年龄范围查询，需要对年龄记录区间和。</td></tr>
<tr><td>时间复杂度</td><td>$O(n)$</td><td>空间复杂度</td><td>$O(1)$</td></tr>
</table>

Problem 10

<table>
<tr><td>试题名称</td><td colspan="3">Sine Partition Function</td></tr>
<tr><td>试题来源</td><td>Codechef OCT 11</td><td>试题编号</td><td>PARSIN</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定n, m, K，问$\sum_k \prod_{i=1}^{m} \sin(k_i K)$，其中$k_i$为自然数。
$m \le 50; n \le 10^9; 0 \le K \le 6.28$，保证答案不超过$10^{300}$，要求精度的绝对或相对误差不超过0.1。</td><td colspan="2">考虑最简单的DP，定义：
$f_{n,m} = \sum_k \prod_{i=1}^{m} \sin(k_i K)$
$g_{n,m} = \sum \cos(k_m K) \prod_{i=1}^{m-1} \sin(k_i K)$
根据合成角公式得到递推式：
$f_{n,m} = (f_{n-1,m-1} + g_{n-1,0}) \sin(K) + f_{n-1,m} \cos(K)$
$g_{n,m} = (f_{n-1,m-1} + g_{n-1,0}) \cos(K) - f_{n-1,m} \sin(K)$
于是使用矩阵乘法就可以优化递推得到答案了。</td></tr>
<tr><td>时间复杂度</td><td>$O(m^3 \log n)$</td><td>空间复杂度</td><td>$O(m^2)$</td></tr>
</table>

Problem 11

试题名称	Colored Domino Tilings and Cutsontest		
试题来源	Codechef NOV 11	试题编号	DOMNOCUT
题目大意		算法讨论	
一个$N*M$列的矩形棋盘。一个棋盘覆盖的染色是指：在棋盘上填上小写字母，使得每个格子有且仅有一个相邻格子的字母和他的一样。一个格子与另一个格子相邻当且仅当他们有公共边。每个字母对应一种颜色。棋盘的割是指一条竖直或水平的直线将棋盘分成两半，这条直线不能穿过一对有相同颜色的相邻格子。现在你需要构造一个在割数尽可能小的情况下，染色数尽可能小的染色。 T为数据组数，$T \le 3000; N, M \le 500$.		NM为奇数时是无解的，设$N \le M$，当$N \le 5$的时候我们手动构造，当$N >= 5$的时候我们可以构造出$5*6$和$6*8$的两种棋盘，这两种棋盘都可以在不改变割数的情况下增加两行或者两列，同时保证可以构造出三染色。于是当N, M较小时打表，否则就构造，就能解决了。	
时间复杂度	$O(TNM)$	空间复杂度	$O(NM)$

Problem 12

试题名称	Luckdays		
试题来源	Codechef NOV 11	试题编号	LUCKYDAY
题目大意		算法讨论	
给定整数A, B, X, Y, Z, P, C，按照以下的方式生成序列S：$S_1 = A, S_2 = B, S_i = (XS_{i-1} + YS_{i-2} + Z) \mod P$。接下来有$Q$组询问，每组询问给出$L_i, R_i$，你需要求出满足$i \in [L_i, R_i]$且$S_i = C$的整数$i$的个数。 T为数据组数，$T \le 2; Q \le 2*10^4; p \le 10007\&prime(p); 1 \le L_i, R_i \le 10^{18}$.		当$XY = 0$时S的循环节不超过P所以可以暴力求解。接下来讨论$XY \ne 0$的情况。 序列的循环节长度可以达到p^2，如果要求S的某一项，可以使用矩阵乘法，矩阵乘法结果是一个三维向量(S_i, S_{i+1}, i)，现在我们要解方程$S_i = C$，就是求向量$(C, x, 1)$的个数，其中$x \in [0, p)$。 此时采用BSGS求解，首先对于0到p^2内所有数，每$\sqrt{p}$个求出它代表的矩阵存到hash 表内，然后枚举x对于每一个可能的矩阵，我们给它依次乘以$\sqrt{p}$次矩阵，并在hash表中查找，这样就能求出第一个循环节内满足$S_i = C$的i。 同理可以求出循环节，有了这两个信息就可以直接算出答案了。	
时间复杂度	$O(Tp\sqrt{p} + TQ\log P)$	空间复杂度	$O(p\sqrt{p})$

Problem 13

<table>
<tr><td>试题名称</td><td colspan="3">Hypertrees</td></tr>
<tr><td>试题来源</td><td>Codechef DEC 11</td><td>试题编号</td><td>HYPER</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">一个3-超图类似与一个普通的图, 只不过其中的边都连接三个点。一个3-超树是一个去掉任意一条边都以后都不连通的3-超图。给定n，问有几种含有n个带标号点的不同的3-超树。
$n \leq 17; ans < 2^{63}$.</td><td colspan="2">对于一棵超树，我们可以把它分成若干个点双连通分量，对于每一个分量都有：去除一个点超树依然连通。可以发现对于一棵点双连通的超树，没一条边连接的三个点都恰好有一个是叶子(除了点数为3 的情况)。我们把这个节点删掉，这样就得到常规的点双连通分量，原来超图的点数对于这个图的点数+边数。
我们暴力搜索点数为i的双连通超树数量，接着我们暴力枚举这个超树的每个点双连通分量的大小，然后记忆话搜索出这些点双拼接出的不同的超树数量。
但是这样程序是会TLE的，本地需要大约7s才能跑出所有的n，于是打表提交即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(T)$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 14

<table>
<tr><td>试题名称</td><td colspan="3">Short II</td></tr>
<tr><td>试题来源</td><td>Codechef DEC 11</td><td>试题编号</td><td>SHORT2</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定质数p，问有多少对$a, b(a > p, b > p)$ 满足ab被$(a-p)(b-p)$整除。
T是数据组数，$T \leq 5; p \leq 10^12$.</td><td colspan="2">可以把原问题等价于求满足$ab|p(a+b+p)$的数对个数。分以下三种情况：
1.a, b同时被p整除，这个时候满足条件的a和b只有5对。
2.a, b同时不被p整除，这是显然$a \neq b$，设$a < b$，这时候满足$a < 1 + \sqrt{p+1}$，此时$b = \frac{a+p}{ak-1}$，其中k是满足条件的任意整数。令$d = ak - 1$，那么一个合法解满足$b > a$，a不被p整除，$d|(a+p)$，且$a|(d+1)$。我们分$b \leq d$和$d \leq b$两种情况枚举较小的数然后检查这些条件，可以发现上界都是$\sqrt{p+1+\sqrt{p+1}}$。就可以计算了。
3.a, b中恰好有一个被p整除，可以发现这种条件的对数是2的两倍，因为对于任意一个满足2条件的数对(a, b)都可以得到恰好两个满足3条件的数对$(a, \frac{p(a+p)}{b})$和$(\frac{p(b+p)}{a}, b)$。</td></tr>
<tr><td>时间复杂度</td><td>$O(T\sqrt{p})$</td><td>空间复杂度</td><td>$O(1)$</td></tr>
</table>

Problem 15

试题名称	Card Shuffle		
试题来源	Codechef JAN 12	试题编号	CARDSHUF
题目大意		算法讨论	
n张卡片，最开始从1到n从上到下摆放着，接下来有m次操作，每一次取出前A张，再取出前B张，然后把前A张放回去再取出前C张，接着把B倒序放回去，最后把C放回。问最后的卡牌顺序。 $n, m \le 10^5$.		直接使用打翻转标记的平衡树模拟即可。	
时间复杂度	$O(m \log n)$	空间复杂度	$O(n)$

Problem 16

试题名称	Misinterpretation 2		
试题来源	Codechef JAN 12	试题编号	MISINT2
题目大意		算法讨论	
一个长度为n的只包含小写字母的字符串S，如果把它的偶数位依次写到开头，再把奇数位依次写下去，得到的字符串和原串一样，那么称这个字符串是好的。给定L, R，问长度为在L, R之间的好的字符串有多少个。 T为数据组数，$T \le 5; L, R \le 10^{10}; R - L \le 5 * 10^4$.		可以发现重排对应一个置换，如果这个置换的循环数为$f(n)$则长度为n的字符串就是$26^{f(n)}$个。当n是偶数的时候，置换后i在$2i \mod n+1$的地方，而n为奇数的时候，第n位不变，所以当$n \mod 2 = 1$时有$f(n) = f(n-1)+1$。于是下面只讨论n为偶数的情况。 令$ord(M)$为2模M的阶，可以发现所有$\gcd(i+1, n) = p$的i每$ord(\frac{n+1}{p})$，一组构成了若干个置换，这样的数有$\phi(\frac{n+1}{p})$个，所以可以得到$f(n) = \sum_{p\|(n+1), p>1} \frac{\phi(p)}{org(p)}$。 考虑怎么求$f(n)$，先求出所有$n$的约数，我们可以对所有$n \in [L-1, R+1]$分解质因数，只要预处理处1到$\sqrt{R}$的所有质数是哪些$n$的约数就可以快速分解$n$了。 接着可以发现当$a, b$互质的时候，$ord(ab) = lcm(ord(a), ord(b))$。所以对质数的幂$p$求出$ord(p)$。这是我们直接枚举$p-1$的约数一一检查来得到答案，当$p \le 10^5$时我们可以预处理，这样对于每一个$n \le [L-1, R+1]$，至多只有一个质数需要重新计算。最后按照式子计算出答案即可。	
时间复杂度	$O(\sqrt{R}\log R + \frac{R^{\frac{3}{4}}}{\log R} + T(R-L)(\frac{\sqrt{R}}{\log R} + \log^2 R))$	空间复杂度	$O(\sqrt{R} + (R-L)\log R)$

Problem 17

试题名称	Find a Subsequence		
试题来源	Codechef FEB 12	试题编号	FINDSEQ
题目大意		算法讨论	
给定一个长度为5的排列p和一个长度为n的整数序列A，现在要求一个A的长度为5的子序列B满足在序列B中恰好有$p_i - 1$个数比B_i小。 T是数据组数，$T \leq 60; n \leq 1000; A_i \in [-10^9, 10^9]$.		枚举第二个数和第四个数的下标，此时可以贪心的选取第一个数和第五个数的值，如果它比第三个数大，就取合法方案中最大的那个，否则就取最小的那个得到这四个数之后就对第三个数的范围和下标有了约数，如果第三个数存在，我们暴力一边得到答案即可。 需要的信息可以通过$O(n^2)$预处理点(i, A_i)的二维前缀和得到。	
时间复杂度	$O(Tn^2)$	空间复杂度	$O(n^2)$

Problem 18

试题名称	Ciel and Earthquake		
试题来源	Codechef MARCH 12	试题编号	CIELQUAK
题目大意		算法讨论	
一个$n * m$的四联通网格图，现在每一条边都有p的概率损毁，问点$(1, 1)$和点(n, m)联通的概率。 T为数据组数，$T \leq 50; n \leq 8; m \leq 10^{18}; 0.1 \leq p \leq 1$.		当m不大的时候我们可以使用轮廓线状态压缩DP，记录轮廓线上每个点的连通性已经是否和$(1, 1)$联通。通过最小表示法，状态数S只有3000多个。预处理转移就可以$O(nmS)$求解了。 当m很大的时候，我们发现答案和m大致呈指数关系即$ans = a * b^m$，我们取足够大的x，例如$x = 40$。则$ans_m = \left(\frac{ans_{x+1}}{ans_x}\right)^{m-x} * ans_x$。	
时间复杂度	$O(T(nxS + \log m))$	空间复杂度	$O(nxS)$

Problem 19

试题名称	Evil Book		
试题来源	Codechef MARCH 12	试题编号	EVILBOOK
题目大意		算法讨论	
有n个人，你打败第i个人需要付出c_i的代价，打败它后可以获得d_i的魔法值，最开始你的魔法值是0。你可以对人使用魔法，对第i个人使用后c_i和d_i都将除以3，魔法可以使用无限多次但是每一次使用要消耗X点魔法值，魔法值不够的之后不能使用魔法。你要使你的魔法值大于等于666，问你最少付出多少的代价。 T表述数据组数，$T \le 5; n \le 10; 10 \le X \le 666$.		除了第一个人之外，战胜每一个问之前都会把它的魔法值除到666以下，如果你用了i次魔法就一定有$iX < \frac{d}{3^i}$，可以得到你对每一个使用魔法的数目范围长度不超过4，所以可以搜索。 可以证明存在一种最优解对每个人使用魔法次数单调不减，于是就能减少很多没用的搜索，再辅助一些其他的剪枝搜索即可。	
时间复杂度	$O(T4^n)$	空间复杂度	$O(n)$

Problem 20

试题名称	Find a special connected block		
试题来源	Codechef APRIL 12	试题编号	CONNECT
题目大意		算法讨论	
一个$n*m$的网格，每一个格子都有一个$[-1, n*m]$范围内的整数权值以及一个代价。问一个代价和最小的四联通块满足联通块中没有权值为-1的格子且至少出现了k种不同的正权。值。$n, m \le 15; k \le 7$.		如果每个格子的权值在$[-1, k]$内，就可以用斯坦纳树解决，$f_{i,j}$表示第i个点为根的颜色集合为j的最小代价和。枚举子集，利用最短路转移即可。 对于$n*m$种权值，我们随意一种$[1, n*m]$到$[1, k]$的映射，对映射之后求斯坦纳树，显然这个解不会更优且有概率和答案一样，和原问题最优解匹配上的概率是$\frac{k!}{k^k}$，我们随机C次取最优解即可。	
时间复杂度	$O(Cnm(3^k + 2^k \log n))$	空间复杂度	$O(nm2^k)$

Problem 21

<table>
<tr><td>试题名称</td><td colspan="3">Substrings on a Tree</td></tr>
<tr><td>试题来源</td><td>Codechef APRIL 12</td><td>试题编号</td><td>TSUBSTR</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一棵n个节点的树，每一个节点都被标记了一个小写字母。一个字符串被这棵树表示了，当且仅当它可以被表示为一个点往它的后代移动路径上的经过所有点的字母连接起来得到的字符串。求有多少个字符串被这颗树表示了。之后有m次询问，每一次询问给出了26个字母的大小顺序，求被这棵树表示的字符串中第K_i小的字符串。
$n \leq 2.5 * 10^5; m \leq 5 * 10^4$输出长度不超过$L = 8 * 10^5$。</td><td colspan="2">对这一棵树建立后缀自动机，按照bfs顺序插入SAM即可，可以证明是$O(n)$的。由于后缀自动机是个DAG，所以第一问只要求出拓扑序倒着DP一遍就行。至于第二问，我们可以预处理从某个点开始的不同的字符串数目，之后dfs一遍就能得到答案了。</td></tr>
<tr><td>时间复杂度</td><td>$O(n+m+L)$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 22

<table>
<tr><td>试题名称</td><td colspan="3">Little Elephant and Boxes</td></tr>
<tr><td>试题来源</td><td>Codechef MAY 12</td><td>试题编号</td><td>LEBOXES</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有n个盒子，第i个盒子里有P_i的概率是V_i块钱，$1 - P_i$ 的概率是一块钻石。你打开了所有盒子之后去买东西，一共有m件物品，第i件需要C_i块钱和D_i个钻石，你一定会买数量尽可能多的物品。问你期望能买到多少件物品。
$n, m, D_i \leq 30; V_i, C_i \leq 10^7$.</td><td colspan="2">首先使用DP预处理买i个物品，有j个钻石，至少需要多少钱，这个可以$O(n^2)$。
使用meet-in-middle方法，枚举前x个盒子的所有情况。记录打开前x个盒子得到a元钱和b个钻石的概率，再按照b放进x个数组中，按照a排序求出前缀和。
再枚举后$n-x$个盒子的情况，对于每一种情况我们枚举一共有多少个钻石及买了多少个物品，推算出前x个物品的情况，再找到对应b的数组，二分出这种情况可行的概率。x取20就可以了。</td></tr>
<tr><td>时间复杂度</td><td>$O(n^3 + x2^x + xn^2 2^{n-x})$</td><td>空间复杂度</td><td>$O(2^x + nm)$</td></tr>
</table>

Problem 23

<table>
<tr><td>试题名称</td><td colspan="3">Selling Tickets</td></tr>
<tr><td>试题来源</td><td>Codechef MAY 12</td><td>试题编号</td><td>TICKETS</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有n道菜肴和m位顾客，每位顾客有两道喜爱的菜，每道菜只能供应给一位顾客。
现在问你最多能招待多少位顾客，使得无论来的是哪些顾客，所有顾客都能够吃到至少一道他喜欢的菜。
$n \leq 200; m \leq 500.$</td><td colspan="2">把菜肴看成点，顾客看成边，设函数$G(E)$表示边集E的邻接点集。如果求出最小的满足$|E| > |G(E)|$的E，答案就是$|E| - 1$。
显然$G(E)$不会有度为1的点否则可以删掉它，而$|E| = |V| + 1$，则只有两种情况：
A.两个点度为3，其余点度为2
B.一个点度为4，其余点度为2
图有两种情况：一是从S到T的三条路径，枚举S到T，bfs求三条最短路径就可以了。二是两个环通过一条链相连，枚举链上的一点，做bfs生成树，则环上有一条边不在树上，枚举这条边，求出环长和深度，去最小值和次小值作为两个环即可。
路径或者环有重叠都可以证明包含更优的子图，所以可以不用排除。</td></tr>
<tr><td>时间复杂度</td><td>$O(n^3 + n^2m)$</td><td>空间复杂度</td><td>$O(n + m)$</td></tr>
</table>

Problem 24

<table>
<tr><td>试题名称</td><td colspan="3">Cool Numbers</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 12</td><td>试题编号</td><td>COOLNUM</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">一个数A有k位，从中选出至多3个不同的数位，令这几位的和为S，这个数的所有数位和为K，如果存在一种选取方案满足$(K = S)^S$是A的倍数，那么就把A称为cool number。现在给定一个n，求小于等于n的最大的cool number和大于n 的最小的cool number。
T为数据组数，$T \leq 10^5, n \leq 10^{1000}; \sum \log_{10}^{n} \leq 10^6.$</td><td colspan="2">我们把cool number分成两类，第一类只有不超过三个非0位，显然这些数都满足条件。大于n最小和小于等于n最大的这一类数都容易求得。
对于第二类，满足$(9k - 27)^{27} > 10^{k-1}$，于是可以得到$k$的上界为77，所以第二类cool number 是有限的，我们预处理出第二类cool number之后每次二分即可。至于预处理，我们可以枚举$K - S$没然后枚举$(K - S)^{27}$的所有因子。可以发现第二类cool number只有不到$N = 4 * 10 * 4$个，但是直接暴力会超时，我们可以放宽条件，不如缩小一点$K - S$的枚举范围，再把剩下的数打表，就可以在时限内做完了。</td></tr>
<tr><td>时间复杂度</td><td>$O(T(\log n + \log N))$</td><td>空间复杂度</td><td>$O(N)$</td></tr>
</table>

Problem 25

试题名称	Expected Maximum Matching		
试题来源	Codechef JUNE 12	试题编号	MATCH
题目大意		算法讨论	
给定一个$n*m$个矩阵$p_{i,j}$，生成一个$n-m$点的二分图，左边第i个点与右边第j个点连边的概率是$p_{i,j}$。求二分图的最大匹配的期望。 $n \le 5; 1 \le m \le 100; 0 \le p_{i,j} \le 1$		根据Hall定理，如果左边的集合S和右边匹配，则每一个S的子集T都要至少与$\|T\|$个右边的点连接。可以使用状态压缩DP，$f_{i,K}$表示只考虑右边前i个点，左边的每一个子集满足上述情形的情况为K，K是一个小于2^{32}的二进制数。可以发现这个条件满足包含关系所以可以通过dfs预处理出所有合法的状态，发现状态数C并不多。所以预处理转移，直接DP即可。	
时间复杂度	$O(Cm)$	空间复杂度	$O(Cm)$

Problem 26

试题名称	Dynamic GCD		
试题来源	Codechef JULY 12	试题编号	DGCD
题目大意		算法讨论	
给定一棵n点带点权树，m个操作： 1.询问两点路径上所有点权的gcd。 2.给一条路径上所有点加一个权值。 $n, m \le 5*10^4$.		首先使用树链剖分转化成序列问题。可以发现$\gcd(a,b,c) = \gcd(a, b-a, c-b)$，所以可以对序列做差分，分别维护每个节点的值与差分后区间的gcd，这可以用线段树实现。查询的时候就计算第一个元素与之后差分序列的gcd的gcd即可。	
时间复杂度	$O(m \log^3 n)$	空间复杂度	$O(n)$

Problem 27

<table>
<tr><td>试题名称</td><td colspan="3">Equivalent Suffix Tries</td></tr>
<tr><td>试题来源</td><td>Codechef JULY 12</td><td>试题编号</td><td>EST</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">T组询问，每组给定一个长度为n的字符串，求由小写字母组成，后缀字母树与该字符串同构的字符串个数 mod 42424242。
$T \le 10; n \le 10^5$.</td><td colspan="2">后缀字母树有如下性质：
1.叶子个数取决于字符串中最长的是其他后缀的前缀的后缀长度，若长度为L则有$N-L$个叶子。
2.与根连接的节点个数取决于原串中所含不同字母的个数。
3.两个后缀的LCA深度取决于LCP的长度。
所以L，字母个数，前$N-L-1$个后缀分别与第$N-L$个后缀的LCP在同构的字符串之间是不变的，取中第三点使得后面一部分字符确定。可以枚举满足限制的前$N-L$个后缀，使得第$N-L+1$个后缀是它的前缀，并利用hash判重，最终字符串的数目就是$K\binom{26}{cnt} * cnt!$，其中K是满足要求的后缀个数，cnt为不同字符数。</td></tr>
<tr><td>时间复杂度</td><td>$O(TN\log N)$</td><td>空间复杂度</td><td>$O(N)$</td></tr>
</table>

Problem 28

<table>
<tr><td>试题名称</td><td colspan="3">A Game of Thrones</td></tr>
<tr><td>试题来源</td><td>Codechef AUG 12</td><td>试题编号</td><td>GTHRONES</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一个序列，序列中有n种数，第i种的权值是u_i，出现了c_i次。两个人博弈轮流操作。第一个回合第一个选取一个数字作为初始的局面值。之后的每一个回合，当前的人需要选出一个和局面值相似的数作为局面的值，之前的局面值将移出游戏(如果有多个只移出一个)，如果无法操作则算输。两个数$a, b(a > b)$是相似的当且仅当$b|a$且$\frac{a}{b}$是质数。你需要判断哪个人必胜，如果是第一个人必胜，你需要输出可以使他获胜的最大的初始局面值是多少。
$n \le 500; u_i \le 10^{18}; c_i \le 10^9$.</td><td colspan="2">如果两个数是相似的就连上边，质数判断使用miller-rabin，根据质因数指数之和的奇偶性分组可以证明是一个二分图。二分图博弈可以使用最大匹配，我们把c_i个点放在一起所以一个点可以匹配c_i条边。如果使用网络流求解那么，第二个人获胜当且仅当存在完备匹配即连接S或者T的边都满流。
可以发现一个数可以作为初始权值，则必定存在一个最大匹配使得这个数没有匹配边。我们可以对每个数做一次退流就可以判断出来了。</td></tr>
<tr><td>时间复杂度</td><td>$O(n^2\log n$
$+maxflow(n, n\log n))$</td><td>空间复杂度</td><td>$O(n\log n)$</td></tr>
</table>

Problem 29

<table>
<tr><td>试题名称</td><td colspan="3">Two Magicians</td></tr>
<tr><td>试题来源</td><td>Codechef AUG 12</td><td>试题编号</td><td>MAGIC</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一张n点m边的简单无向图，两个人博弈，最开始两个人分别在1号点和2号点。从第一个人开始轮流操作，每一个回合有以下三个步骤：
1. 可以沿着现有的无向边移动任意步，如果这一步结束时两个人在同一个格子，则当前人胜。
2.加入一条连接两个节点现在还没有的无向边，如果无法加入则另一个人胜。
3. 最开始每一个人有P次传送机会，如果当前人还有传送的机会，他可以选择消耗一次并传送到任意一个节点。问谁必胜。
T表示数据组数，$T \leq 100; n \leq 7777; m, p \leq 10^4$.</td><td colspan="2">使用DP，需要记录的状态如下：两人所在联通块的奇偶性，两人剩下的传送次数，奇数块的个数，偶数块的个数，添加之后不会影响连通性的边的数目的奇偶性。枚举操作就可以转移了，这样复杂度是$O(n^2P^2)$。
考虑优化，观察并使用程序验证介意发现如下规律：传送最多使用一次，当奇数块足够多的时候DP值以4为周期循环，当偶数块足够多的时候DP值不变，所以状态数就是$O(1)$的了。
所以先DP预处理然后每次求出相应数值带入即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(Tm\alpha(n))$</td><td>空间复杂度</td><td>$O(n+m)$</td></tr>
</table>

Problem 30

<table>
<tr><td>试题名称</td><td colspan="3">Knight Moving</td></tr>
<tr><td>试题来源</td><td>Codechef SEPT 12</td><td>试题编号</td><td>KNGHTMOV</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">一个无限大的棋盘，一开始有一个骑士在$(0,0)$。骑士有两种运动方式：$(+A_x, +A_y)$和$(+B_x, +B_y)$。同时棋盘上还有K个障碍物。问有多少种方法把骑士移动到坐标(X, Y)，对10^9+7取模，如果有无穷多种方案就输出-1。
$K \leq 15$，其余数字绝对值不超过$D = 500$。</td><td colspan="2">如果(A_x, A_y)和(B_x, B_y)线性无关，则每个可以写成$\alpha A + \beta B$的位置都可以映射到(α, β)。不考虑障碍物的话就是$\binom{\alpha+\beta}{\alpha}$，有障碍物就容斥一下。
如果(A_x, A_y)和(B_x, B_y)线性相关，问题就变成了一维问题。可以发现和问题有关的坐标范围只有$[-D^2, D^2]$，把每个点映射到数轴上然后暴力DP即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(K^2 2^K + D^2)$</td><td>空间复杂度</td><td>$O(D^2)$</td></tr>
</table>

Problem 31

<table>
<tr><td>试题名称</td><td colspan="3">Annual Parade</td></tr>
<tr><td>试题来源</td><td>Codechef SEPT 12</td><td>试题编号</td><td>PARADE</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一张n个点的有向图，有若干个英雄在魔法道路上旅行，对于每一名英雄，他从城市$begin_i$开始，途径一些城市，最终在城市end_i结束。注意，$begin_i$可能等于end_i，但他在路径中必须至少向另外的一个城市移动。他可以多次经过一条道路，但每次都需要付出费用。
游行的费用分成三部分：
1.沿道路移动的总费用。（如果一条道路被k人次经过，费用就将计算k次）
2.如果一名英雄的$begin_i \neq end_i$，需要花费C来送他回家。
3.如果某个城市没有被任何一名英雄经过，需要花费C来赔偿该市市民。
C的值每年都可能改变，而我们知道接下来K年的这个值。你的任务是，计算每一年的最小花费。
$N \leq 250; 1 \leq M \leq 30000; 1 \leq K \leq 10000; 1 \leq V_i, C_i \leq 10000.$</td><td colspan="2">首先我们可以用Floyd求出任意两点之间的最短路，接着就可以直接考虑英雄经过哪些关键点了，关键点之间的距离直接使用最短路$g_{i,j}$即可。
考虑使用费用流解决。类似最小路径覆盖，我们把一个点拆成两个组成一个二分图，对于每对顶点(i,j) 我们从左边的第i个点连向右边第j个点，容量为1费用为$g_{i,j}$。
接下来我们考虑送英雄回家的花费和补偿市民的花费，发现费用恰好为$(n-flow)C$，$flow$是流量。于是我们可以只再当前网络流图上最短路不超过C时增广，就能得到最优答案了。
由于C在不断变化，我们可以离线处理，将C从小到大排序，这样逐步放松增广的限制，一步步加大流量，就能在相当于一次费用流的时间内解出所有的C的答案了。</td></tr>
<tr><td>时间复杂度</td><td>$O(maxflow(N,N^2)$
$+K\log K)$</td><td>空间复杂度</td><td>$O(N^2+K)$</td></tr>
</table>

Problem 32

试题名称	Max Circumference		
试题来源	Codechef OCT 12	试题编号	MAXCIR
题目大意		算法讨论	
给定二维平面中的三个点ABC和N个操作，第i个操作有两个参数x_i, y_i，使用这个操作可以使得点A的x坐标增加x_i，y坐标增加y_i。你可以使用最多K个操作，每一个操作最多使用一次，ABC三个点允许共线或者重合。现在你需要最大化$\|AB\| + \|BC\| + \|AC\|$，答案的绝对误差必须小于10^{-12}。 $K \leq N \leq 500; \|x\|, \|y\| \leq 10^9, \|x_i\|, \|y_i\| \leq 10^6$.		问题可以转化成最大化以B, C为焦点的过A的椭圆，考虑过A最优解的切线，则一定存在u, v使得最大化$uA_x + vA_y$就能的到最优解。 如果已经知道了u, v按照每个操作的贡献排序取前K个正贡献操作即可，考虑当(u, v)连续变化的时候，只有在$ux_i + vy_i = 0$或者$ux_i + vy_i = ux_j + vy_j$的关键点出我们的决策才可能出现变化。所以我们只要从某一个位置顺时针出发然后更新操作的贡献顺序，同时维护x_i, y_i的前缀和，每次二分得到前K个中最后一个正数的位置，直接算出当前答案更新最优解即可。 注意开方直接调用sqrt()的精度不够，设$I = \lfloor \sqrt{S} \rfloor, D = \sqrt{S} - I$。则使用公式$D = \frac{S-I^2}{I+\sqrt{S}}$就能保证精度了。	
时间复杂度	$O(N^2 \log N)$	空间复杂度	$O(N^2)$

Problem 33

试题名称	Arithmetic Progressions		
试题来源	Codechef NOV 12	试题编号	COUNTARI
题目大意		算法讨论	
给定长度为N的数列A，统计这样的三元组数量：对于$1 \leq i < j < k \leq N$，有$A_j - A_i = A_k - A_j$。 $N \leq 10^5; A_i \leq 30000$		考虑分块，块大小B，分成S块。对于每个块，枚举i, j或者j, k在块内的情况，用桶计算可行的k或者i的数量。 对于只有j在块内的情况，我们可以对式子化简得到$A_i + A_k = 2A_j$。对两边的桶做卷积，在枚举A_j，就可以统计了。最后用FFT优化卷积即可。 时间复杂度$O(B * B * S + S * C \log C)$，调节块大小可以做到$O(n\sqrt{n \log n})$。	
时间复杂度	$O(n\sqrt{n \log n})$	空间复杂度	$O(n + C)$

Problem 34

试题名称	Martial Arts		
试题来源	Codechef NOV 12	试题编号	MARTARTS
题目大意		算法讨论	
一个完全二分图，边有两个权值$A_{i,j}$和$B_{i,j}$，要进行匹配。令匹配边的A值总和为H，B值总和为G。 对手的目的是最大化$G-H$，其次最大化G，他会在知道了匹配之后选择是否去掉一条匹配边，使得该边的权值不算入H和G。 任务是找一个完全匹配，最大化$H-G$，其次最大化H。 $n \le 100; A_{i,j}, B_{i,j} \le 10^{12}$.		定义权值$W_{i,j} = A_{i,j} - B_{i,j}$。 对手删边当且仅当这条边最大且为正值，KM算法不能直接解决。我们考虑枚举对手会删除哪一条边，按照关键字从小到大枚举，对于对手来说删掉它总是最优的，所以就变成了加入一条边，强制匹配上这一条边，询问最大匹配。 我们首先把边权全部变成$-\infty$，然后加入一条边并强制匹配就是修改为$+\infty$，再修改回来。于是只要处理更改一条边权之后维护最大匹配。 使用KM算法中每个点的$Label$值，假设改变(i,j)的权值，我们将i和它之前的匹配点删掉，然后维护$Label_i$，令$Label_i = \max_j(W_{i,j} - Label_j)$，再从$i$点出发找一条增广路使得匹配完全。这样就不必重构整个图了。这样每次维护是$O(n^2)$的。	
时间复杂度	$O(N^4)$	空间复杂度	$O(N^2)$

Problem 35

试题名称	Different Trips		
试题来源	Codechef DEC 12	试题编号	DIFTRIP
题目大意		算法讨论	
给定一棵n个顶点的数，每个节点的权值定义为它的度数，两条路径相同当且仅当长度相同且对应位置权值相同。徐闻这棵树有多少条不同的从孩子走向祖先的路径。 $n \le 10^5$.		直接仿照SA的求法，把倍增改成树上倍增，如果我们把每次求SA的rank数组保留下来，就得到了一个SA的倍增数组，这样求任意两个点的LCP就可以直接通过树上倍增在$O(\log n)$内完成。 我们倍增k次，当$2^k \ge n$时我们把SA数组拿出来用每个节点的深度减去相邻两个节点的LCP就是答案。	
时间复杂度	$O(n \log n)$	空间复杂度	$O(n \log n)$

Problem 36

试题名称	A New Door		
试题来源	Codechef JAN 13	试题编号	ANDOOR
题目大意		算法讨论	
有一个全黑的二维平面，将$(0,0)$和(W,H)确定的矩形涂成白色。再给定N和圆，涂成黑色，求白色部分的周长，只计算**严格**在矩形内部的部分。 多组数据：$N \le 1000; \sum N \le 5000$ $n, m \le 10^5; a, b \le 1000$		显然一段周长一定是某个圆周的一部分，我们枚举每个圆，统计该圆合法的周长总长，对于每个圆枚举其他圆和矩形的四条边，求出覆盖的周长的极角区间，排序后统计，只要支持圆和圆，圆和直线求交点就可以了。注意精度。	
时间复杂度	$O(n^2 \log n)$	空间复杂度	$O(n)$

Problem 37

试题名称	Cucumber Boy and Cucumber Girl		
试题来源	Codechef JAN 13	试题编号	CUCUMBER
题目大意		算法讨论	
给定B个$n*n$的矩阵A_i，对于数对$(a,b)(a < b)$定义$n*n$的矩阵B满足$B_{i,j} = \sum_{k=1}^{n} A_{a,i,k} * A_{b,j,k}$，一个$1-n$的排列$P$是好的当且仅当至少存在一个$i$使得$B_{i,p_i}$是奇数，数对$(a,b)$是好的当且仅当好的排列有奇数个。 询问有多少好数对。 $n \le 60; B \le 8000.$		对于矩阵B，令矩阵$C_{i,j} = B_{i,j} + 1 \mod 2$。考虑$C$的行列式，每个不好的排列的贡献都是$\pm 1$，而好的排列的贡献都是0,。当$n > 1$时数对$(a,b)$是好的等价于$det(C)$是奇数。 因为$C_{i,j} = (\sum_{k=1}^{n} A_{a,i,k} * A_{b,j,k}) + 1 \mod 2$。所以可以在每个矩阵后面补上全是1的第$n+1$列，那么$C = A_a * A_b \mod 2$。 考虑$det(C)$，根据Binet-Cauchy定理，令$A_{i,j}$为矩阵$A$删掉第$j$列后得到的矩阵，那么就有$det(C) = \sum_{i=1}^{n+1} det(A_{a,i}) + det(A_{b,i})$。所以只要求出所有的$det(A_{a,i})$即可。如果$A_i$满秩，则可以吧$A$消成$n*n$的单位阵加上一列的形式，设为$D$，假设加上的是第$k$列，则$det(A_{i,j}) = D_{j,k}(j < k), det(A_{i,k}) = 1, det(A_{i,j}) = 0(j > k)$。 于是直接统计即可，消元统计部分可以用bitset优化。	
时间复杂度	$O(n^2 B + B^2)$	空间复杂度	$O(n^2 B)$

Problem 38

<table>
<tr><td>试题名称</td><td colspan="3">Observing the Tree</td></tr>
<tr><td>试题来源</td><td>Codechef FEB 13</td><td>试题编号</td><td>QUERY</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">维护n个点的无根树上的点权，三种操作：
1.给结点x到y路径加等差数列，首项为a，公差为b；
2.询问结点x到y路径上的点权和；
3.回到第x次修改操作之后的状态。
操作数量为m。强制在线。
$n, m \leq 10^5; a, b \leq 1000$</td><td colspan="2">区间数据结构维护：
标记可以用k, b表示为区间加上$k * i + b$。
由于满足交换性，可以不下传标记。
直接使用可持久化线段树维护树链剖分即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(m \log^2 n)$</td><td>空间复杂度</td><td>$O(m \log^2 n)$</td></tr>
</table>

Problem 39

<table>
<tr><td>试题名称</td><td colspan="3">Room Corner</td></tr>
<tr><td>试题来源</td><td>Codechef FEB 13</td><td>试题编号</td><td>ROC</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一个$n * m$的网格图，用$+, -, |$描述了一个房间的边界，保证房间内的所有格子都是四联通的且同一行的所有房间内的格子相邻。现在这个房间所有90度的内角对应的格子中都站了一个小朋友，小朋友移动的时候把手放那个内角所对应的墙上，且每时每刻手不能离开墙壁。我们把90度内角所对应的空白格成为特殊点。接着游戏开始，每时每刻两个相邻的(即之间没有任何小朋友) 都处在特殊点上的小朋友可以交换位置，它们都向对方的特殊点移动且这次交换直到双方同时到达对方的特殊点才终止。小朋友移动到相邻的空白格需要一个单位的时间。现在有T组询问，每组询问问在最优情况下两个小盆友相遇至少需要多少时间。
$T \leq 10^4; n, m \leq 2500$.</td><td colspan="2">首先求出所有关键点沿着墙逆时针到出发点的距离，然后考虑两个小朋友的交换步骤，一定是相向交换，有两个方向可以选择，对于一个方向，我们可以二分出它们路径的终点，就能算出从这一侧交换所需的时间，取最小值就是答案了。</td></tr>
<tr><td>时间复杂度</td><td>$O(nm + T \log n)$</td><td>空间复杂度</td><td>$O(nm)$</td></tr>
</table>

Problem 40

试题名称	Little Elephant and Colored Coins		
试题来源	Codechef MARCH 13	试题编号	LECOINS
题目大意		算法讨论	
给定n种给定面值V_i 和颜色C_i的硬币，每种硬币都有无穷多个。接下来有Q组询问，你要选出一些硬币使得它们的和为S，需要最小化选出硬币中的颜色种类数，无解输出-1。 $1 \le n \le 30; 1 \le V_i, Q \le 2 * 10^5; 1 \le S \le 10^{18}$.		取出最小的面值m，设$f_{i,j}$表示前i种颜色的硬币可以拼成的模m为j的最小面值，然后转移时特判作为基准的硬币对应的颜色即可。	
时间复杂度	$O(n^2V)$	空间复杂度	$O(nV)$

Problem 41

试题名称	String Query		
试题来源	Codechef APRIL 13	试题编号	STRQUERY
题目大意		算法讨论	
一个长度为10的只包含小写字母的字符串S，n个操作： 1. 在首或者尾或者正中间插入一个小写字符。 2. 在首或者尾或者正中间删除一个小写字符，保证字符串长度不会小于10。 3. 给定一个字符串T，问T在S中出现了多少次。 $n \le 1.5 * 10^5; \sum \lvert T \rvert \le 1.5 * 10^6$.		如果只在一端插入删除可以使用后缀平衡树，如果只在首位删除，我们可以维护两棵后缀平衡树，分别维护一个前缀和后缀，查询时分别在两边查询之后再从中间开始两边各取出$\lvert T \rvert - 1$个字符使用KMP暴力判断。当两个后缀平衡树中有一个为空时，把字符串均分重建出两棵后缀平衡树，这样复杂度是均摊$O(n \log n)$的。 如果要支持在中间的插入删除操作的话。我们就可以把字符串均分成两段，两边都按照上面的方法维护起来，插入删除的时候都实时维护从中间分开的分界点的位置，每次维护是$O(\log n)$ 的。查询时也类似上面的方法，先在两边按照上面的方法查询，然后再从中间开始两边都取出$\lvert T \rvert - 1$个字符进行KMP 暴力匹配即可。 于是维护4个后缀平衡树就能支持所有操作了。	
时间复杂度	$O((n + \sum \lvert T \rvert) \log n)$	空间复杂度	$O(n + \lvert T \rvert)$

Problem 42

<table>
<tr><td>试题名称</td><td colspan="3">Queries on tree again!</td></tr>
<tr><td>试题来源</td><td>Codechef MAY 13</td><td>试题编号</td><td>QTREE</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一棵n个节点的带边权的基环外向树，两点之间的最短路径定义为经过点数最少的路径。保证环的大小是奇数，即保证任意两点的最短路径唯一。
接下来进行m次操作：
1. 把两点之间最短路径上的所有边的权值取相反数。
2. 询问两点之间最短路径上的边权的最大子段和。
$1 \leq n, m \leq 10^5$.</td><td colspan="2">我们把环断开一条边$< root, spc >$，一个点作为根，另一个点作为特殊点，找(u, v)最短路径时，我们可以枚举三种情况，取最短路径即可：$u \to v$，$u \to root \to spc \to v$，$u \to spc \to root \to v$。
采用树链剖分的话，就转化成了序列上的问题。
用线段树维护的话，维护区间和，区间内答案，左/右端点开始最大/小子段和就可以维护了。注意信息是有顺序的。</td></tr>
<tr><td>时间复杂度</td><td>$O(n \log^2 n)$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 43

<table>
<tr><td>试题名称</td><td colspan="3">Count Special Matrices</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 13</td><td>试题编号</td><td>SPMATRIX</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">$N(N >= 3)$是一个正整数。A是一个$N * N$的整数矩阵。这个矩阵如果满足以下条件就称它是特殊的：
$A_{x,x} = 0 for 1 \leq x \leq N$
$A_{x,y} = A_{y,x} > 0, 1 \leq x < y \leq N$
$A_{x,y} \leq max(A_{x,z}, A_{z,y}), 1 \leq x, y, z \leq N$
$A_{x,y} \in \{1, 2, \cdots, N-2\}, 1 \leq x < y \leq N$
任意$k \in \{1, 2, \cdots, N-2\}$ 存在$x, y \in \{1, 2, \cdots, N\}$ 满足$A_{x,y} = k$.
求大小为$N * N$的特殊矩阵的个数，模$10^9 + 7$。
T为数据组数，$T \leq 10^5; N \leq 10^7$.</td><td colspan="2">可以得到答案是$\frac{n!(n-1)!}{3*2^{n-1}}\left(\frac{3}{2}n - 2 - H_{n-1}\right)$。
其中$H_n = \sum_{k=1}^{n} \frac{1}{k}$。
具体证明见我的第一部分作业详细题解。
所以我们直接预处理调和数，2的方幂，阶乘和阶乘逆元，对于每个询问就能$O(1)$的算出答案了。</td></tr>
<tr><td>时间复杂度</td><td>$O(T + N)$</td><td>空间复杂度</td><td>$O(N)$</td></tr>
</table>

Problem 44

试题名称	Two k-Convex Polygons		
试题来源	Codechef JUNE 13	试题编号	TKCONVEX
题目大意		算法讨论	
给定n个棍子的长度和整数k，求能否在其中选出$2k$个棍子拼成两个凸多边形。使得两个凸多边形都恰好有k根棍子组成,且任意相邻的边都不共线。 $n \le 1000; k \le 10; length \le 10^9$.		由于边长有限制，事实上当n超过70的时候一定有解，我们可以排序之后暴力一遍得到一个多边形，删掉之后再暴力一遍得到第二个多边形，就可以了。 当$n < 70$的时候，我们分两种情况：其中一个多边形的最小边大于另一个多边形的最大边，这个还用上述的方法就可以判定。否则如果有解，则一定是排序之后连续$2k$根棍子。 我们直接枚举是哪一段$2k$根棍子，然后直接枚举每根棍子分配给哪一个多边形，然后在判断是否可行即可。	
时间复杂度	$O(n \log n + 70\binom{2k}{k})$	空间复杂度	$O(n)$

Problem 45

试题名称	Music & Lyrics		
试题来源	Codechef AUG 13	试题编号	LYRC
题目大意		算法讨论	
给定n个字符串S和m个字符串T，询问每个S在T中一共出现多少次。 $n \le 500; \|S\| \le 5000; m \le 100; \|T\| \le 50000$.		对所有的S串建立AC自动机，然后所有的T跑一遍，匹配上一个点就把权值+1。 最后每个S的出现次数就是fail树的子树和。	
时间复杂度	$O(n\|S\| + m\|T\|)$	空间复杂度	$O(n\|S\| + \|T\|)$

Problem 46

试题名称	Prime Distance On Tree		
试题来源	Codechef AUG 13	试题编号	PRIMEDST
题目大意		算法讨论	
给一颗树，边权为1，随机选两个点，求距离为质数的概率。 $2 \le n \le 50000$.		显然只要统计长度为质数的路径数目即可，使用经典的树分治套FFT 解决。从当前根开始按照子树大小升序dfs，$d_{x,i}$表示第x棵子树中深度为i的节点个数，计算对答案的贡献只要把d_x和$sd_x = \sum_{y<x} d_y$做FFT即可。 NTT要注意模数不能过小。	
时间复杂度	$O(n \log^2 n)$	空间复杂度	$O(n)$

Problem 47

试题名称	Two Roads		
试题来源	Codechef SEPT 13	试题编号	TWOROADS
题目大意		算法讨论	
平面上给定n个点，现在你要选择两条直线，最小化所有节点到这两条直线上最近点的路径的平方和。保证不存在重点和三点共线。$n \le 100$.		可以发现两条直线的两条垂直的角平分线把平面分成了四个区域。每半条直线控制一个区域。我们枚举控制区域的划分，就可以通过线性回归得到答案。我们首先枚举一条角平分线，可以发现一定是某两个点的连线，确定了一个角平分线之后，另一条角平分线的划分方案数是$O(n)$的，我们按照第一条角平分线重新心意x坐标再扫描线即可。需要维护的量有：$\sum x, \sum y, \sum x^2, \sum y^2, \sum xy$。 这样我们共枚举$O(n^3)$种序列划分，然后根据线性回归以及点到直线距离公式，可以算出答案，更新即可，由于每个点会自动选取最近的路，所以线性回归得到的直线即使不满足区域划分也不影响最优解的选取。	
时间复杂度	$O(n^3 \log n)$	空间复杂度	$O(n)$

Problem 48

试题名称	Fibonacci Number		
试题来源	Codechef OCT 13	试题编号	FN
题目大意		算法讨论	
给定M, C询问满足$fib_n \mod M = C$的最小的n，无解输出-1。T为数据组数，$T \le 1000; 0 \le C < M \le 2*10^9$，$M$为质数且$M \mod 10$为完全平方数。		先写出斐波那契数列模意义下的通项公式，问题就变成了解方程$p^n + p^{-n} = a$，可以解出$p^n = \frac{a \pm \sqrt{a^2+4}}{2}$，如果$\sqrt{a^2+4}$有解，就可以使用BSGS求$n$，否则无解，注意要分$n$的奇偶性讨论。模意义下开方可以使用Cipolla’s algorithm。	
时间复杂度	$O(\sqrt{P})$	空间复杂度	$O(\sqrt{P})$

Problem 49

试题名称	Gangsters of Treeland		
试题来源	Codechef NOV 13	试题编号	MONOPLOY
题目大意		算法讨论	
给定一棵n个节点的有根树，开始每一个点都有一个不同的权值，接下来进行m次操作： 1. 把i到根路径上的所有节点标记成一种新的权值。 2. 询问i子树中所有节点到根路径上不同权值个数的平均值。 $n, m \le 2*10^5$.		修改操作类似LCT的access的操作，可以证明偏爱子节点的切换次数是均摊$O(n \log n)$的。 同时我们维护每个节点到根的路径上非偏爱路径的边数。 我们用LCT维护每条链，每次操作时access一下，切换偏爱子节点时注意维护一下原来和现在的偏爱子节点子树的权值变化，使用DFS序+树状数组维护即可。	
时间复杂度	$O(Q\log^2 n)$	空间复杂度	$O(n)$

Problem 50

试题名称	Queries With Points		
试题来源	Codechef NOV 13	试题编号	QPOINT
题目大意		算法讨论	
平面上给出n个简单K_i边形，接着给出Q次询问，每次询问给定一个点在哪个多边形内，或者不在任何多边形内，强制在线。 $n, Q \le 10^5; \sum K_i \le 3*10^5; \lvert x\rvert, \lvert y\rvert \le 10^9$.		如果没有强制在线，直接使用经典点定位算法就行了： 使用扫描线，把任意一条线段的端点x坐标当做关键点，关键点之间的区域，多边形的边的相对顺序是不变的，我们用平衡树维护线段，我们只要对一个点求出在它上方或者下方的线段，然后判断这两条线段是不是同一个多边形的上边界和下边界即可。 如果强制在线的话，我们在进行扫描线时使用可持久化平衡树记录线段，对于每个点，我们二分出对于x坐标的区间，找到对应的平衡树版本，在其中查询即可。	
时间复杂度	$O((K+Q)\log K)$	空间复杂度	$O(K\log K)$

Problem 51

试题名称	Query on a tree VI		
试题来源	Codechef DEC 13	试题编号	QTREE6
题目大意		算法讨论	
给一棵n个节点的数，每一个节点都是白色或者黑色。你需要支持两个操作： 1.修改一个点的颜色。 2.如果只保留相同颜色点之间的边，询问一个点所在联通块的大小。 $n, m \le 10^5$.		对黑点和白点分别维护一个树链剖分，对于每个节点维护一下只考虑同色节点的联通子树大小。 对于每次修改的节点x，我们求出从x**的父亲**开始往上，找到第一个**不在集合里**的节点y，然后把$fa(x) \to y$(包括y不包括x)的路径上更新x节点的贡献(加入就是加，删除就是减)。使用树状数组就可以维护同色联通子树大小了，对于一个询问我们就找到$x \to y$ 的y之前的一个节点，输出它的同色节点的联通快大小即可。 寻找y的节点的方法：我们可以用树状数组维护一段dfs区间内的节点数目，我们一条重链一条重链的往上找，知道发现某一条重链上从当前点到顶端有断点，这时候再二分确定断点位置，复杂度是$O(n \log^2 n)$ 的。	
时间复杂度	$O(m \log^2 n)$	空间复杂度	$O(n)$

Problem 52

试题名称	Petya and Sequence		
试题来源	Codechef DEC 13	试题编号	REALSET
题目大意		算法讨论	
给定一个长度为n的整数序列A，问是否存在一个不全为0的整数序列B满足对于所有的$0 \le j < n$ 都有 $\sum_{i=0}^{n-1} A_i * B_{i+j \mod n} = 0$。 $T \le 100; n \le 3 * 10^4$.		问题等价于判定矩阵$X_{i,j} = A_{i+j \mod n}$是否满秩。 令$f(x) = \sum_{i=0}^{n-1} A_i x^i$。如果$gcd(f(x), x^n - 1)$的次数为$d$，那么这个矩阵的秩就是$n - d$。 定义$\phi_n(x)$为$\prod_{d\|n} \phi_d(x) = x^n - 1$，那么$\phi_x(x)$无法表示为两个次数均不为0 的多项式的乘积。所以只要判断是否存在$d\|n$满足$\phi_d(x)\|f(x)$。这个问题等价于$f(x) = \prod_{i\|d\&prime(i)}(x^{\frac{d}{i}} - 1)$ 被$x^d - 1$整除。 直接模拟即可。	
时间复杂度	$O(n \log n + ndiv(n))$	空间复杂度	$O(n)$

Problem 53

试题名称	Counting D-sets		
试题来源	Codechef JAN 14	试题编号	CNTDSETS
题目大意		算法讨论	
在n维空间中，两点距离定义为每一维坐标差的绝对值的最大值，点集的直径定义为距离最远的两点的距离，两个点集相同当且仅当它们可以通过平移得到。求n维空间中直径等于D的点集个数。 $n \le 10^3; D \le 10^9$		我们只要求出直径不大于D的点集数目$ans(D)$，$ans(D) - ans(D-1)$即是答案。平移的问题我们可以考虑令每一维都有一个点的该维坐标为0。考虑容斥，枚举k为不能取0的维数。 $ans(n) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} 2^{D^k (D+1) n-k}$	
时间复杂度	$O(n^2)$	空间复杂度	$O(n^2)$

Problem 54

试题名称	Counting The Important Pairs		
试题来源	Codechef JAN 14	试题编号	TAPAIR
题目大意		算法讨论	
给定一张n个点m条边的简单无向图，求有多少种删掉恰好两条边的方案使得这张图不连通。 $n \le 10^5; m \le 3 * 10^5$		建立一棵生成树，每一条非树边赋一个随机权值，树边的权值是覆盖它的所有非树边权值的异或和。当随机范围足够大时，删去一个边集能使图不连通的重要条件是存在一个子集权值异或和为0。由于只删除两条边，排序扫一遍统计即可。	
时间复杂度	$O(m \log m + n)$	空间复杂度	$O(n+m)$

Problem 55

试题名称	Count on a Treap		
试题来源	Codechef FEB 14	试题编号	COT5
题目大意		算法讨论	
一个初始为空的max-treap，现在进行n次操作： 1.插入一个给定权值和优先级的几点。 2.删除一个节点。 3.询问(u, v)之间的路径长度。 保证权值和优先级两两不同。 $n \le 2 * 10^5$.		两个节点的LCA就是他们对应区间中优先级最高的点，于是LCA很好求，问题就变成了求一个点的深度。 考虑节点y，y是x的祖先当且仅当x和y之间不存在优先级比y大的节点。考虑使用线段树来维护一个节点左侧(右侧)的祖先数目。设函数$x.getl(c)$表示在线段树上节点x当前最大值是c时的祖先个数，我们要维护当前区间的最大值，和在左子树中询问右子树最大值的结果，这样一个询问就能被递归到左右两边之一。同理可以解决$x.getr(c)$。这样维护节点信息就是$O(n \log n)$ 的，于是线段树维护信息和查询的复杂度就变成$O(n \log^2 n)$了。	
时间复杂度	$O(n \log^2 m)$	空间复杂度	$O(n)$

Problem 56

<table>
<tr><td>试题名称</td><td colspan="3">Graph Challenge</td></tr>
<tr><td>试题来源</td><td>Codechef FEB 14</td><td>试题编号</td><td>DAGCH</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一个n个m边的图的DFS序，每一个点的标号为它的DFS序。、保证所有节点都能从1号点到达。一个节点x对y来说是好的当且仅当$x < y$ 且存在一条x到y的路径使得中间节点编号都大于y。一个节点x对y来说是最好的当且仅当它是所有对y的好节点中编号最小的。给定Q个询问，每个询问问对于一个节点，它是多少个节点的最好的节点。
$1 \leq n, Q \leq 10^5; n - 1 \leq m \leq 2 * 10^5$.</td><td colspan="2">根据定义，可以发现一个节点最好的节点就是dominator tree算法中的半必经点，直接运行dominator tree算法再统计即可。</td></tr>
<tr><td>时间复杂度</td><td>$O((n + m)\alpha(n))$</td><td>空间复杂度</td><td>$O(n + m)$</td></tr>
</table>

Problem 57

<table>
<tr><td>试题名称</td><td colspan="3">Chef and Graph Querie</td></tr>
<tr><td>试题来源</td><td>Codechef MARCH 14</td><td>试题编号</td><td>GERALD07</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一张n点m边的无向图，给出Q个询问$[L_i, R_i]$，对于每个询问你需要输出如果只保留编号区间$[L_i, Ri]$中的边，图中有多少个联通块。
$n, m, Q \leq 2 * 10^5$.</td><td colspan="2">我们按照编号顺序加入每条边，用LCT维护编号最大生成树。那么当前加入到第r条边时询问$[l, r]$的答案就是$n - c$，c表示当前最大生成树中编号大于等于l的边的数目。</td></tr>
<tr><td>时间复杂度</td><td>$O((m + Q) \log n)$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 58

<table>
<tr><td>试题名称</td><td colspan="3">The Street</td></tr>
<tr><td>试题来源</td><td>Codechef MARCH 14</td><td>试题编号</td><td>STREETTA</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定长度为n的数列A,B，刚开始A全部为0，B全部为$-\infty$。支持三种共m个操作：
1.对A区间加一个等差数列。
2.对B区间对一个等差数列取max。
3.求A_i+B_i。
$n \le 10^9; m \le 3*10^5$.</td><td colspan="2">区间加等差数列很好维护，只要记录首项和公差就可以了，标记可以永久化。
区间等差数列取max我们已经采用永久化的标记，问题就在于如何合并线段树节点上的两个标记。
等差数列可以看成直线，那么两条直线最多只有一个交点，如果没有交点，保留较大的那一条即可。否则我们求出交点的位置，如果交点在mid的右侧，我们就在当前节点上存储上左半边比较大的那个等差数列，然后把另一个等差数列下传到有孩子继续合并到线段树中；交点在mid左侧的情况类似。下传到叶子节点就可以停止了。</td></tr>
<tr><td>时间复杂度</td><td>$O(m\log^2 n)$</td><td>空间复杂度</td><td>$O(m\log n)$</td></tr>
</table>

Problem 59

<table>
<tr><td>试题名称</td><td colspan="3">Chef and Tree Game</td></tr>
<tr><td>试题来源</td><td>Codechef APRIL 14</td><td>试题编号</td><td>GERALD08</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有一棵n节点的树，树上每一条边的颜色都是红色或者蓝色。现在两个人轮流操作，第一个人每次选择一条红色边删除，第二个人每次选择一条蓝色边删除。
每次删除之后与1号点不连通的点和边都会被删除。不能操作的人输。
两个人都使用最优策略，求第一个人和第二个人先手时分别谁赢得游戏。
$n \le 10^5$</td><td colspan="2">这是不平等博弈，可以使用超现实数解决。对每个节点定义一个局面函数f：
叶子节点函数值为0。每个节点的函数值为所有孩子的贡献之和。对于每个孩子c如果连接它和它父亲的是红边，则令a为$f_i+a>1$的最小正整数，它的贡献是$\frac{f_i+a}{2^{a-1}}$。如果连接它和它父亲的是蓝边，则令a为$f_i-a>-1$的最小正整数，它的贡献是$\frac{f_i-a}{2^{a-1}}$。
如果根节点函数值是0，那么谁后手谁赢，如果是整数，无论如何第一个人赢，否则无论如何第二个人赢。
至于高精度小树的维护，可以发现一个大小为s的子树的函数值小数位数不超过s，于是开一棵平衡树维护二进制小数中所有的1，采用启发式合并来实现加法。</td></tr>
<tr><td>时间复杂度</td><td>$O(n\log^2 n)$</td><td>空间复杂度</td><td>$O(n\log n)$</td></tr>
</table>

Problem 60

试题名称	Dynamic Trees and Queries		
试题来源	Codechef MAY 14	试题编号	ANUDTQ
题目大意		算法讨论	
给一棵n个点的带点权的树(树根为1)，m个操作，操作有以下四种。 1. 加入一个给定点权的叶子。 2. 删除一个子树。 3. 给一个子树中的所有节点的权值加上一个值。 4. 询问一个子树中的权值和。 $n, m \leq 10^5$		LCT很难直接维护子树，考虑转换问题，我们在每个节点上维护子树大小siz和当前子树和sum以及子树上每个节点要加的值add。计算x答案时我们求x到根的路径和来计算x的子树中每个点应当加上多少，再加上x的子树本身权值和就可以了。即$siz * dta + sum$。 加入叶子时要先求一下add以去除之前的操作对当前节点的影响。删除子树时维护siz和sum即可，子树加时除了给当前点打上dta的单点标记之外还要给到根的链上统一加上$siz * d$，d是加上的数。 这三个信息都可以通过链上修改单点求值或者单点修改链求和完成，所以就可以使用LCT维护了。	
时间复杂度	$O(m \log n)$	空间复杂度	$O(n)$

Problem 61

试题名称	Sereja and Subsegment Increasings		
试题来源	Codechef MAY 14	试题编号	SEINC
题目大意		算法讨论	
给两个长度为n的序列A, B，满足$0 \leq A_i, B_i < 4$，每一次选择一个区间$[l, r]$，把A数组的区间$[l, r]$加一再模4，蚊最少进行多少次操作后能把数组A变成数组B。 $n \leq 10^5$.		我们令$C_i = B_i - A_i \mod 4$，再令$D_i = C_i - C_{i-1} \mod 4$操作就变成了把$D_l$减1，把$D_{r+1}$加1，要让$D$数组全部变成0，我们采用以下贪心策略。 首先忽略掉0然后扫一遍，把前面的1与后面的3匹配，只用一次操作就消掉两个数字，可以证明没有更优的方法。 接下来序列就变成了2中间隔排列着若干个3和若干个1，3一定在1之前。 从前往后枚举每个3，若它不是第一个元素，就把它+1，把第一个元素-1。我们再从后往前枚举每个1，若它不是最后一个元素，就把它-1，把最后一个元素+1。 这样每个1或者3只使用一次就消去了，接着整个序列除了首尾两个元素不确定之外其余的都是2，我们不停的把首元素-1，尾元素+1，这样消完即可，显然不存在更优的做法。	
时间复杂度	$O(n)$	空间复杂度	$O(n)$

Problem 62

<table>
<tr><td>试题名称</td><td colspan="3">Sereja and Arcs</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 14</td><td>试题编号</td><td>SEAARC</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">一个长度为n的序列A，对于每一对$(i,j)(i < j)$，若满足$A_i = A_j$，就以$(i,0)(j,0)$为直径画圆，颜色为A_i。问有多少对颜色不同的圆存在交点。$1 \le n, A_i \le 10^5$</td><td colspan="2">数对的关系只能是$ABAB, AABB, ABBA$，考虑用总数减去$AABB, ABBA$两种。
$AABB$很好计算，求出右端点为1到i的圆的个数与左端点为i的圆的个数的乘积。
下面考虑$ABBA$，可以分两种情况，令$S = \sqrt{\frac{n}{\log n}}$：
1.两种颜色中有一种数个数超过S，就可以通过求出这种颜色左侧(右侧)右侧有多少个这种颜色的数，然后把每个小区间的贡献加起来得到答案。这一部分的复杂度是$O(n\sqrt{n\log n})$。
2.两部分个数都不超过S，可以发现满足这个条件产生的弧的个数是nS的，直接把这nS个弧按照(i,j)放到平面上通过扫描线计算数目即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(n\sqrt{n\log n})$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 63

<table>
<tr><td>试题名称</td><td colspan="3">Two Companies</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 14</td><td>试题编号</td><td>TWOCOMP</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一棵树以及树上链的集合A, B，集合中每条链都有权值，你需要分别选出A, B的一个子集使得任意一个树上的顶点不同时被两个子集中的链覆盖。
$n \le 10^5; |A|, |B| \le 700$.</td><td colspan="2">显然这是非常经典的二分图最大匹配的问题。问题就在一如何判断两条链相交。
我们可以把一条路径$u \to v$拆分成两个部分，即$u \to LCA(u,v)$和$v \to LCA(u,v)$。这两条链都是竖直的，
判断两条竖直的链是否相交可以设上方顶点较浅的链为$x \to y$，另一条是$p \to q$，如果p在x的子树内且y在p的子树内，就说明相交。</td></tr>
<tr><td>时间复杂度</td><td>$O(n\log n + maxflow(|A| + |B|, |A||B|))$</td><td>空间复杂度</td><td>$O(n + |A||B|)$</td></tr>
</table>

Problem 64

<table>
<tr><td>试题名称</td><td colspan="3">Game of Numbers</td></tr>
<tr><td>试题来源</td><td>Codechef JULY 14</td><td>试题编号</td><td>GNUM</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给两个长度为n的数组A, B，每次选择两个数对$(i,j),(p,q)$，满足$A_i < B_j, A_p < B_q$。$gcd(A-i, B_j, A_p, B_q) > 1$，$(i,j)$不在集合$S_1$中，$(p,q)$不在集合$S_2$中。然后把$(i,j)$加入$S_1$，把$(p,q)$加入$S_2$。问最多进行多少轮游戏。
$n \le 400.$</td><td colspan="2">把每一个$gcd(x,y) > 1$的数对拿出来，根据x, y的大小关系可以分成两部分，如果两个属于不同部分的数不互质，就连边，于是这张图的最小割就是答案。但是这样边数太多，是$O(n^4)$ 的，需要优化。
对于每一个出现过的质数p，我们建立一个点，(x,y)向$gcd(x,y)$的所有质因数连边即可，这样的图和原图等价。而边数只有$O(n^2 \log n)$。</td></tr>
<tr><td>时间复杂度</td><td>$O(maxflow(n^2, n^2 \log n))$</td><td>空间复杂度</td><td>$O(n^2 \log n)$</td></tr>
</table>

Problem 65

<table>
<tr><td>试题名称</td><td colspan="3">Sereja and Equality</td></tr>
<tr><td>试题来源</td><td>Codechef JULY 14</td><td>试题编号</td><td>SEAEQ</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">两个序列相似当且仅当元素相对顺序相同。
对于两个排列P_1, P_2，定义$F(P_1, P_2)$表示这样的区间$[l, r]$个数：
$P_1[l, r]$与$P_2[l, r]$相似，且$P_1[l, r]$逆序对数不超过E。
给出n, E求：
$\sum F(p_1, p_2)[P_1, P_2 \in perm(n)]$。
数据组数$10000; n \le 500; E \le 10^6$.</td><td colspan="2">枚举$[l, r]$区间长度$x$，考虑剩下的$n - x$个元素填写顺序显然有$(\frac{n!}{x!})^2$ 种，区间的位置有$n - x + 1$种，区间内元素相对顺序本质不同的排列共有$\sum_{i=0}^{E} cnt(x, i)$ 种。
所以答案就是$\sum_{x=1}^{n} \sum_{i=0}^{E} cnt(x, i)$
$cnt(x, i)$表示逆序对数为i的x的排列数，这个可以DP预处理：
枚举$x + 1$插入序列中的位置，共$x + 1$种情况，可以采用前缀和优化，这部分复杂度是$O(n^3)$。
每个询问枚举x暴力算就行了，注意常数优化。</td></tr>
<tr><td>时间复杂度</td><td>$O(n^3 + Tn)$</td><td>空间复杂度</td><td>$O(n^3)$</td></tr>
</table>

Problem 66

试题名称	Team Sigma and Fibonacci		
试题来源	Codechef AUG 14	试题编号	SIGFIB
题目大意		算法讨论	
给定n,m，求$\sum(6xyz*fib_x*fib_y*fib_z*[x+y+z=n]) \mod m$ $n \le 10^{18}; m \le 10^5$.		可以推出这个数列的母函数是$f(x) = 6g(x)x^3(x^2+1)^3$ 其中$[x^n]g(x) = \frac{1}{3125}[25\binom{n+5}{5}(fib_{n+6}+2fib_{n+5}) + 150\binom{n+4}{4}fib_{n+5} + 5\binom{7}{2}\binom{n+3}{3}(fib_{n+4}+2fib_{n+3}) + 5\binom{8}{3}\binom{n+2}{2}fib_{n+3} + \binom{9}{4}(n+1)(fib_{n+2}+2fib_{n+1}) + \binom{10}{5}fib_{n+1}]$ 求组合数的时候暴力约分，斐波拉契数使用矩阵快速幂计算，就可以直接计算整个式子了。	
时间复杂度	$O(\log n)$	空间复杂度	$O(1)$

Problem 67

试题名称	Fibonacci Numbers on Tree		
试题来源	Codechef SEPT 14	试题编号	FIBTREE
题目大意		算法讨论	
给定一棵n个节点的树，你要支持以下操作： 1.给路径$u \to v$加上斐波拉契数列的第i项。 2.询问以x为根时y的子树和。 3.询问路径$x \to y$的权值和。 4.让整棵树回到第i次操作以后的状态。 所有答案对10^9+9取模。 $n, Q \le 10^5$.		我们首先解决区间加斐波拉契数的问题。 首先利用特征根法求出斐波拉契数的通项公式：$fib_i = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^i - (\frac{1-\sqrt{5}}{2})^i]$。 而$\sqrt{5}$在模$10^9+9$意义下是存在的数，于是就是区间加两种公比的等比数列。 这显然是可以合并同类项从而实现线段树打标记的，标记就是整个区间加的等比数列的首项。等比数列的和与每一项都可以预处理出来。 于是就解决了区间加斐波拉契的问题。 至于原题，我们直接用可持久化线段树维护树的轻重链剖分即可，注意下传标记时要新建孩子。	
时间复杂度	$O(n+Q\log^2 n)$	空间复杂度	$O(Q\log^2 n)$

Problem 68

<table>
<tr><td>试题名称</td><td colspan="3">Rectangle Query</td></tr>
<tr><td>试题来源</td><td>Codechef SEPT 14</td><td>试题编号</td><td>QRECT</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">在二维平面中支持三种操作：
1.插入一个矩形
2.删除一个矩形
3.给一个矩形，询问当前有多少个矩形和这个矩形有公共点。
Q表示操作数，$Q \le 10^5$.</td><td colspan="2">首先对操作序列CDQ分治，问题就变成了先插入若干矩形，接着询问若干矩形询问有交点的矩形数目，
统计矩形可以采用扫描线的方法，注意与区间$[l, r]$相交区间$[a, b]$的数目写成：区间$[a, b]$的总数减去满足$b < l$的区间数再减去满足$a > r$的区间数，
使用两个树状数组维护当前状态统计答案即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(Q\log^2 Q)$</td><td>空间复杂度</td><td>$O(Q)$</td></tr>
</table>

Problem 69

<table>
<tr><td>试题名称</td><td colspan="3">Union on Tree</td></tr>
<tr><td>试题来源</td><td>Codechef OCT 14</td><td>试题编号</td><td>BTREE</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一棵n个节点的树，树的每一条边的长度都是1。接下来有Q组询问，每组询问给出了K_i个数对(a_i, r_i)，表示距离点a_i在r_i以内的所有点都被守护了，问一共有多少点被守护了。
$n, Q \le 5 * 10^4; \sum K_i \le 5 * 10^5$.</td><td colspan="2">首先考虑给定一组a, r，求距离a不超过r的节点数，这个可以通过点分治预处理来得到答案。对于每一个重心，维护这个重心代表的那个联通块中，重心的每个子树中距离这个重心距离不超过k 的点数，可以开n 个vector记录，总复杂度是$O(n\log n)$的。然后对于一个(a, r)的查询，依次向上一级一级的找重心然后统计当前点所在子树之外的其他子树的贡献，加起来就可以了，所以一次查询$count(a, r)$是$O(\log n)$的。
接下来考虑原问题，我们可以对输入的K个点建立一棵虚树。然后如果x到y距离是d而$r_x - d > r_y$，就可以用$r_x - d$更新r_y，类似最短路算法把虚树更新一遍。就可以直接分每条边讨论了。每条边(x, y)的贡献就是$count(x, r_x) + count(y, r_y) - count(z, d)$，其中$z$是$(x, y)$上$x$ 和y对它控制度相同的点之一，d就是x, y到z的剩余控制距离，可以证明这个z一定存在且仅存在一个。为了使z在整点上，可以在原树中每条边上加一个点。</td></tr>
<tr><td>时间复杂度</td><td>$O(n\log n + \sum K\log n)$</td><td>空间复杂度</td><td>$O(n\log n)$</td></tr>
</table>

Problem 70

试题名称	Children Trips		
试题来源	Codechef OCT 14	试题编号	TRIPS
题目大意		算法讨论	
给定一棵n个点的树，树上每条边的长度都是1或者2。 接下来Q组询问，每组询问表示一个人从u走到v他每天最多走d的长度，且每一天的终止点必须是树上的点，问需要走几天。 $n, Q \le 10^5$.		我们根据d把询问分成两个部分即$d <= \sqrt{n}$和$d > \sqrt{n}$。 对于$d > \sqrt{n}$的询问，答案不超过$O(\sqrt{n})$，我们直接在树上利用倍增数组和dep数组进行模拟即可。 对于$d <= \sqrt{n}$的询问，我们把d相同的一起处理，预处理出向上走一天能走到哪里，向上走2^k天能走到哪里，就可以$O(\log n)$处理每个询问了。	
时间复杂度	$O(n\sqrt{n}\log n)$	空间复杂度	$O(n\log n)$

Problem 71

试题名称	Chef and Churu		
试题来源	Codechef NOV 14	试题编号	FNCS
题目大意		算法讨论	
给定一个长度为n的数组A和n个函数$f_i = \sum_{j=L_i}^{R_i} A_j$。 要求支持两种操作：单点修改$A_i$的值和询问$\sum_(i = l)^r f_i$。 m为操作数。 $n, m \le 10^5$		首先我们考虑定期重建，每$O(\sqrt{m})$次操作重新计算一次A_i和f_i，利用前缀和数组，这部分复杂度是$O(n\sqrt{m})$的。 我们对函数序列分块，预处理每个A_i对每一整块$O(\sqrt{n})$个函数的贡献。这样每次修改A_i我们可以在$O(\sqrt{n})$复杂度内更新每一块函数的总和。 对于每个询问，整块的暴力，多余部分可以使用树状数组暴力算函数值。这样询问操作是$O(n\sqrt{n}\log n)$的。	
时间复杂度	$O(n\sqrt{m} + n\sqrt{n}\log n)$	空间复杂度	$O(n\sqrt{n})$

Problem 72

试题名称	Sereja and Order		
试题来源	Codechef NOV 14	试题编号	SEAORD
题目大意		算法讨论	
有n个程序和两台电脑，第i个程序在第一台电脑上要运行A_i秒，在第二台电脑上运行B_i秒，现在要在最少的时间内完成所有程序在两台电脑上的运行任务，求出最少时间并输出一个方案。 $n \le 10^5$.		答案的下界是$\max(\sum A_i, \sum B_i, A_i + B_i)$。可以证明这个下界一定能取到。如果下界取在$A_i + B_i$就直接把剩下的值填进去。否则一定有一台计算机在不间断的运行。可以发现最优解的顺序可以有很多种，考虑随机一个顺序判断是否可行，采用贪心得到另一台计算机的运行时间下限，判断是否等于答案。是的话我们就得到了一个解。	
时间复杂度	$O(n*$玄学$)$	空间复杂度	$O(n)$

Problem 73

<table>
<tr><td>试题名称</td><td colspan="3">Divide or die</td></tr>
<tr><td>试题来源</td><td>Codechef DEC 14</td><td>试题编号</td><td>DIVIDEN</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">平面上给定一个n度角，你需要通过尺规作图来把这个角分割成n的大小为1度的角。你的操作有：
1. 以A, B为给顶点画一条直线。
2. 以A为圆心，BC之间距离为半径话一个圆。
你用到的点只能是最开始给定的三个点或者你绘制的图形之间的交点。
$0 < n < 360$，坐标范围不超过1000，操作次数不得超过1000。</td><td colspan="2">当$n \mod 3 = 0$的时候无解，其他情况都有解，我们先画一个正五边形，接着画一个正三角形，用72度减去60度就得到了12度，平分两次就得到了3度角，显然$\gcd(3, n) = 1$所以我们只要先从给定角的一边开始都箱内不停的划分3度角，之后再从另一边得到一个1度角，用这个1度角不停划分即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(n)$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 74

<table>
<tr><td>试题名称</td><td colspan="3">Course Selection</td></tr>
<tr><td>试题来源</td><td>Codechef DEC 14</td><td>试题编号</td><td>RIN</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">有n门课，m个学期，每个学期可以修多门课，每门课必须在某一个学期内学，第i门课在第j个学期学习的收益为$X_{i,j}$，若$X_{i,j} = -1$表示第j个学期不开设第i门课。
除此之外还有K个前置关系a, b表示第b门课必须在第a门课之后。
最大化所有课的平均收益。
$1 \le n, m \le 100; 0 \le k \le 100; -1 \le X_{i,j} \le 100$</td><td colspan="2">把问题转化成最小割：
把第i门课拆成$m+1$个点$I_{i,0} \cdots I_{i,m}$。
在$I_{i,j}$和$I_{i,j+1}$之间连边为$max_{1 \le j \le m}\{X_{i,j}\} - X_{i,j}$。（注意要把$-1$变成$-\infty$）
对于前置关系a, b，我们只要对于任意$0 \le j \le m-1$，把$I_{a,j}$和$I_{b,j+1}$连$+\infty$的边。
最后用$X_{i,j}$每行的最大值的总和减去最小割就可以了。</td></tr>
<tr><td>时间复杂度</td><td>$O(maxflow(n, nm))$</td><td>空间复杂度</td><td>$O(nm)$</td></tr>
</table>

Problem 75

<table>
<tr><td>试题名称</td><td colspan="3">Ranka</td></tr>
<tr><td>试题来源</td><td>Codechef JAN 15</td><td>试题编号</td><td>RANKA</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定围棋的规则和$9*9$的棋盘，允许不走，要求棋盘状态和当前先手玩家的状态不能重复，要求设计一种走棋方案，能走n步合法的操作。
$n \leq 10^4$</td><td colspan="2">我们讲斜线依次编号，则每个点(x,y)所在斜线的编号为$x+y-1$。
黑字摆第一条斜线，白子摆第二条，黑子不动，白子摆第一条，黑子摆第三条，白子摆第一条，黑子摆第二条，白子不动，黑子摆第一条，白子摆第四条—……
可以这么写：
work(x)
  if ($x=1$) 摆第1排;
  else
    work($x-1$);
    摆第x排
    if ($xmod2=0$)
      放弃一轮
    work(x-1);
这样据说能走400000步，当然很容易的过了。</td></tr>
<tr><td>时间复杂度</td><td>$O(n)$</td><td>空间复杂度</td><td>$O(\log n)$</td></tr>
</table>

Problem 76

<table>
<tr><td>试题名称</td><td colspan="3">Xor Queries</td></tr>
<tr><td>试题来源</td><td>Codechef JAN 15</td><td>试题编号</td><td>XRQRS</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一个初始时为空的整数序列(元素由1开始标号)以及一些询问：
A.在数组最后加入数字x。
B.在区间$[L,R]$中找到数字y，最大化$(x\ xor\ y)$,xor表示按位异或。
C.删除数组最后k个元素。
D.在区间L,R中，统计小于等于x的元素个数。
E.在区间L,R中，找到第k小的数(第k个顺序统计量)。
$1 \leq M \leq 5*10^5; a_i \in [0,2^{31})$</td><td colspan="2">我们对序列建立前缀和可持久化trie树，按照二进制从高位到低位为关键字，维护子树元素个数。这样寻找y的时候我们只要保证子树内有元素的同时尽量往x相反的方向走就可以了。会发现这个trie也就是一颗动态开节点的值域线段树，于是区间和和第k大也能轻松解决。</td></tr>
<tr><td>时间复杂度</td><td>$O(n \log a)$</td><td>空间复杂度</td><td>$O(n \log a)$</td></tr>
</table>

Problem 77

<table>
<tr><td>试题名称</td><td colspan="3">Payton numbers</td></tr>
<tr><td>试题来源</td><td>Codechef FEB 15</td><td>试题编号</td><td>CUSTPRIM</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">定义三元组(a,b,c)的乘法运算，其中$c=11$或者$c=24$。
如果令$(a1,b1,c1)$和$(a2,b2,c2)$相乘：令$s=(a1a2+b1b2+c1c2)+(a1b2+b1a2)+(c1+c2)$;
$t=\lfloor\frac{s}{2}\rfloor+16(c1+c2)-c1c2$
$A=(t-2(a1b2+b1a2)-(a1c2-c1a2)+33(a1+a2)+(b1b2-a1a2))$
$B=(t-5(a1b2+b1a2)-(c1b2+b1c2)+33(b1+b2)+(2b1b2+4a1a2))$
如果s是偶数，那么结果就是$(A-540,B-540,24)$，否则结果就是$(A-533,B-533,11)$。
定义单位元e满足$e*A=A$，零元素$zero$满足$zero*A=zero$。
一个三元组是素数当且仅当不存在两个非单位非零三元组相乘能得到它。
给一个三元组，判断它是否是素数。
$T\le 10^4; a,b\in[-10^7,10^7]$.</td><td colspan="2">令$\omega$为方程$\omega^2-\omega+3=0$的根即$\omega=\frac{1+\sqrt{-11}}{2}$，那么对于每个三元组$(a,b,c)$，都有到域$Z[\omega]$的映射$\phi(a,b,c)\to(33-2a-c)+(b-a)\omega$。问题就变成了判断域$Z[\omega]$下$a+b\omega$是否为素数。定义共轭$(a+b\omega)'=(a+b-b\omega)$，令$N(x)=xx'$，有如下结论：
1.若x不是整数，那么x是质数当且仅当$N(x)$是质数。
3.若x是整数，那么x是质数当且仅当x是质数且满足$|x|=2$或者$|x|\ne 11$且-11在模x意义下没有二次剩余。
直接使用欧拉判别法和miller-rabin判断即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(T\log|a|)$</td><td>空间复杂度</td><td>$O(1)$</td></tr>
</table>

Problem 78

<table>
<tr><td>试题名称</td><td colspan="3">Devu and Locks</td></tr>
<tr><td>试题来源</td><td>Codechef FEB 15</td><td>试题编号</td><td>DEVLOCK</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">对于所有的$0\le m\le M$，求出所有满足各位数字之和不超过m且是P的倍数的N位数的个数。可以有前导0。$n\le 10^9;(p\le 50;M\le 500)OR(p\le 16;M\le 15000)$.</td><td colspan="2">设$f_{i,j,k}$表示各位数字之和为$k$且模$P$为$j$的$i$位整数个数，直接DP太慢了，我们考虑倍增：
$f_{i1,j1,k1}*f_{i2,j2,k3}\to f_{i1+i2,j1*10^i2+j2\ mod\ P,k1+k2}$
发现最后一维的DP是卷积的形式，我们可以把$f_{i,j,k}$看成多项式$f_{i,j}(x)$。这样就能使用FFT优化了。</td></tr>
<tr><td>时间复杂度</td><td>$O((P+\log M)PM\log n)$</td><td>空间复杂度</td><td>$O(PM)$</td></tr>
</table>

Problem 79

<table>
<tr><td>试题名称</td><td colspan="3">Random Number Generator</td></tr>
<tr><td>试题来源</td><td>Codechef MARCH 15</td><td>试题编号</td><td>RNG</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一个K次线性递推式$A_i = \sum_{j=1}^{K} C_j A_{i-j} \ mod\ 104857601$，现在已知$A_i, C_i (1 \le i \le K)$，求$A_n$。
$n \le 10^{18}; K \le 3 * 10^4$.</td><td colspan="2">这是一个经典问题，我们可以采用多项式取模来优化线性递推式。
具体的，这个线性递推矩阵的他正多项式为$f(x) = x^K - \sum_{i=1}^{K} C_i x^{K-i}$。
我们直接用x^n对$f(x)$取模得到$g(x)$，就能求出A_n在$A_i (1 \le i \le K)$的系数。
于是$\sum_{i=1}^{K} [x^{i-1}] g(x) A_i$就是答案。
求模$f(x)$意义下x^n，可以使用快速幂，同时使用多项式取模。</td></tr>
<tr><td>时间复杂度</td><td>$O(K \log K \log n)$</td><td>空间复杂度</td><td>$O(K)$</td></tr>
</table>

Problem 80

<table>
<tr><td>试题名称</td><td colspan="3">Counting on a Tree</td></tr>
<tr><td>试题来源</td><td>Codechef MARCH 15</td><td>试题编号</td><td>TREECNT2</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一棵n个节点带边权的树，Q次操作，每一次操作修改一条边权。第一次操作之前以及每次操作后输出这棵树中有多少条权值为1的路径，路径权值定义为路径上边权的gcd。
设Z为边权最大值，$1 \le n \le 10^5; 0 \le Q \le 100; 1 \le Z \le 10^6$</td><td colspan="2">将询问离线处理，设$g(k)$表示权值为k的路径数目，显然$ans = \sum_{k=1}^{Z} \mu(k) * g(k)$。
枚举$c \in [1, Z]$，计算$k = c$时对答案的贡献，这时我们可以只考虑这样的边：存在某一时刻，边权为c的倍数。枚举这些边的总复杂度是$2^{cnt(w)}$，其中w为边权，$cnt(x)$表示x的不同质因子数。采用并查集维护路径数，先把不变的边权加入并查集中，对于每一个询问，暴力加入当前时刻剩余边的权值，加入之后要还原，采用启发式合并即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(2^k (n + Q^2) \log n + Z)$，$k$表示最多不同质因子数</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Problem 81

试题名称	Black-white Board Game		
试题来源	Codechef APRIL 15	试题编号	BWGAME
题目大意		算法讨论	
一个$n * n$的01矩阵A，第i行的第L_i到第R_i的格子为1，其他都是0。 两个人玩游戏，每一轮游戏他们同时报出一个还未出现的排列P，满足A_{i,P_i}为1，且第一个人排列的逆序对数为奇数，第二个人的为偶数。 如果有一轮一个人找不到排列就输了，同时找不到就算平局。 给定n, L_i, R_i，判断游戏结果。$n \leq 10^5$		逆序对数为偶数的排列设为1，奇数的设为-1。则根据定义偶排列数目减去奇排列的数目就是A的行列式。 只要求出$\vert A\vert$的符号就能判断游戏结果。 直接使用高斯消元是$O(n^3)$的，考虑使用数据结构优化。 我们建立n棵左偏树，初始把i以R_i为关键字插入第L_i棵左偏树。然后我们从大到小遍历所有左偏树，如果当前左偏树为空，则说明这一列全部为0，行列式必定为0。否则我们把R_i最小的元素i取出来，然后把这棵左偏树合并到第R_i+1棵左偏树中去。这样就完成了消元。 注意要维护元素编号，最后求一次逆序对来确定符号。	
时间复杂度	$O(n \log n)$	空间复杂度	$O(n)$

Problem 82

<table>
<tr><td>试题名称</td><td colspan="3">Make It Zero 3</td></tr>
<tr><td>试题来源</td><td>Codechef APRIL 15</td><td>试题编号</td><td>LMATRIX3</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">使用给定方法生成一个长度为N的序列B，每次操作可以任意选择数组的一个区间$[L,R]$然后吧$B_i(i \in [L,R])$变成$B_i+k \mod P$，问多少次才能把数组全部变成0.
$N=\sum_{i=1}^{m} L_i; m \le 100; P \le 19; L_i \le 10^15$.</td><td colspan="2">我们首先把数组差分得到x，然后就可以转化为把x分成若干个子序列使得每个子序列是P 的倍数，对于每个长度为n的子序列我们都用$n-1$次操作把它变成0。，所以最后操作次数是$n+1-k$，其中k是子序列个数。
然后差分序列只跟每个元素出现次数有关，这样只要存P个数就行了。首先$i+j=P$则一定把i,j凑成一组。这样匹配之后剩下至多五种数字，且如果有5中至少有一种只有一个。我们可以预处理所有可能在答案中的子序列，这个子序列不满足它的一个子集是P的倍数，这样的子序列并不多，只有30种。
我们可以随便选出一个初始解，然后考虑优化这个解，每次优化一定是把一个子序列分裂成更多的子序列，接着发现不能被其他优化线性表出的优化只有130个，每个优化出现次数不超过12，所以可以使用DP直接求出最优的优化方案然后暴力搜索得到最优解。
由于给定的数组每一段的循环节都很小，所以可以直接暴力循环节得到。</td></tr>
<tr><td>时间复杂度</td><td>$O(mP+$玄学$)$</td><td>空间复杂度</td><td>$O(m+P)$</td></tr>
</table>

Problem 83

<table>
<tr><td>试题名称</td><td colspan="3">Little Party</td></tr>
<tr><td>试题来源</td><td>Codechef APRIL 15</td><td>试题编号</td><td>LPARTY</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定m个不同的长度为n的01串，一个基子集可以用一个数列S和一个长度为—S—的01串s表示，一个基子集能覆盖的01串A满足对于任意的$1 \leq i \leq |S|$，都有$si = A_{S_i}$。现在你需要使用一些基子集，使得所有给定的串都可以被这些基子集覆盖且没有给定的串都没有被这些基子集覆盖。你需要最小化使用的基子集的大小的和。
T为数据组数，$T \leq 120; m \leq 1000; n \leq 5$.</td><td colspan="2">这是一个最小带权集合覆盖问题，是NPC的。只能考虑对搜索进行优化，我们先枚举所有基子集，判断是否可行，然后把可行的存下来，数目不会超过3^n，于是直接搜索复杂度是$O(2^{2^n})$，显然会TLE，需要优化。
如果一个集子集能被另一个集子集完全包含，那么这个集子集就没必要选，这样去掉一些之后最多只有32个集子集。在搜索的过程中，如果当前的总和超过最优解或者剩下的完全选上都不能完全覆盖，就可以进行剪枝，这样就能在时限内解决了。</td></tr>
<tr><td>时间复杂度</td><td>$O(T2^{3^n})$</td><td>空间复杂度</td><td>$O(3^n + m)$</td></tr>
</table>

Problem 84

试题名称	Chef and Balanced Strings		
试题来源	Codechef MAY 15	试题编号	CBAL
题目大意		算法讨论	
一个字符串是平衡的当且仅当它的每个字符都出现了偶数次。一个字符串的$type$权值是它所有平衡字串长度的$type$次方和。给一个长度为n的字符串，有Q个询问$L, R, type$询问字串$[L, R]$的$type$权值。$n, Q \leq 10^5; type \in \{0, 1, 2\}$.		用一个二进制数表示每个字符在前缀中的出现次数的奇偶性，讲字符串变成一个数组A。$[L, R]$是平衡的当且仅当$A_{L-1} = A_R$。我们可以首先对A离散化，这样值域就在n以内了。 接着我们考虑如何暴力计算权值，我们枚举起点，把终点往后扫，开一个桶维护每种权值出现的次数，位置和，位置平方和。这样我们加入一个元素的时候就能$O(1)$算出以该元素为结尾的平衡字串的权值和。 接着我们对A数组分块。我们可以预处理$Pre_{i,j}$表示第i块到位置j的权值和，$Suf_{i,j}$表示位置i到第j块的权值和，$Val_{i,j}$表示块i到块j的权值和。 对于一个询问我们分成三段，块p的后缀，一段完整的块，块q的前缀。 则答案分成两部分，即起点终点都不在完整的块里的部分X，和有一个在完整的块里的部分Y。 对于X由于序列总长度不超过$2\sqrt{n}$我们可以暴力。 对于Y我们可以分别计算包括左右两边再减去公共部分：$Y = Pre_{p+1,R} + Suf_{L,q-1} - Val_{p+1,q-1}$。 于是$X + Y$就是答案。	
时间复杂度	$O(n\sqrt{n})$	空间复杂度	$O(n\sqrt{n})$

Problem 85

试题名称	Counting on a directed graph		
试题来源	Codechef MAY 15	试题编号	GRAPHCNT
题目大意		算法讨论	
给定一张n点m边的有向图一个点对(X, Y)合法当且仅当存在一条从1到X的路径和一条从1到Y的路径，除了1之外没有公共点。 统计无序点对(X, Y)的数目。 $n \leq 10^5; m \leq 5 * 10^5$.		可以发现(X, Y)合法当且仅当X和Y没有除1之外的公共必经点。 先求出dominator tree，然后条件就变成了X和Y的LCA为1，于是扫描1的所有子树计算答案即可。	
时间复杂度	$O((n + m)\alpha(n))$	空间复杂度	$O(n + m)$

Problem 86

<table>
<tr><td>试题名称</td><td colspan="3">Chefbook</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 15</td><td>试题编号</td><td>CHEFBOOK</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定m个限制a_i, b_i, W_i, L_i, R_i，你要设定n个非负整数X_i和n个非负整数Y_i。满足对于每一个限制都有$L_i \leq X_{a_i} - Y_{b_i} + W_i \leq R_i$。你需要最小化$\sum_{i=1}^{m} X_{a_i} - Y_{b_i} + W_i$并输出一种最优方案。
$n \leq 100; m \leq n^2$.</td><td colspan="2">把限制写成$x_i - x_j \leq K$的形式，对于每个限制，在i, j之间链上一条流量无限，费用为K的边。如果X_i在限制中出现了k边，就从S到X_i连一条容量为k的边，Y_j同理。求最小费用最大流就是答案。
但是这个答案是原问题对偶得到的，可以使用差分约束构造一个解，对于图中每条流量不为0 的边，说明这个限制刚好被达到，于是所有流量不为0的边都加入差分约束系统求一个可行解就是合法的最优解。</td></tr>
<tr><td>时间复杂度</td><td>$O(maxflow(n, m), nm^2)$</td><td>空间复杂度</td><td>$O(m)$</td></tr>
</table>

Problem 87

<table>
<tr><td>试题名称</td><td colspan="3">Easy exam</td></tr>
<tr><td>试题来源</td><td>Codechef JULY 15</td><td>试题编号</td><td>EASYEX</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">一个K面的骰子，投到每一面的概率相同，现在你投n 次，设投完之后数字i出现了a_i 次，试求$\sum_{i-1}^{L} a_i^F$的期望。
$n, K \leq 10^9; 1 \leq L \leq K; 0 \leq L * F \leq 5 * 10^5; 1 \leq F \leq 1000$</td><td colspan="2">设$x_{i,j}$表示第$j$次是否投到$i$，显然$x_{i,j}$的期望是$\frac{1}{K}$。答案要求$\prod_{i=1}^{L}(\sum_{j=1}^{n} x_{i,j})^F$。把式子展开之后我们考虑每一项，一项中若存在$a, b, c(a \neq b)$使得$x_{a,c}$和$x_{b,c}$的次数都不为0，显然这一项没有意义，贡献为0.否则这一项中每个变量都是独立的，设p表示无关变量的个数，期望就是$\frac{1}{K^p}$。
既然一项的贡献无关变量的数目p有关，我们不妨计算有i个无关变量的贡献不为0的项的系数和。设$w_{i,j}$表示$(\sum x_k)^i$展开之后含有j 个无关变量的项的系数和。
枚举新加入的自变量可以得到递推式：
$w_{i,j} = w_{i-1,j-1} + j * w_{i-1,j}$
令$f_{n,m}$表示$\prod_{i=1}^{n}(\sum_{j=1}^{n} x_{i,j})^F$中出现了$m$ 个无关变量的项的系数之和，那么久有：
$f_{n,m} = f_{n-1,j} * w_{F,m-j} * \binom{m}{j}$
发现递推式是卷积的形式，所以可以使用FFT+快速幂优化，这样就得到了贡献相同的每一组项的系数和，最后统计答案即可。</td></tr>
<tr><td>时间复杂度</td><td>$O(LF \log LF \log L)$</td><td>空间复杂度</td><td>$O(LF)$</td></tr>
</table>

Problem 88

试题名称	A game on a graph		
试题来源	Codechef JULY 15	试题编号	HAMILG
题目大意		算法讨论	
给定一张无向图，两个人玩游戏。第一个选择一个出发点，接着第二个人开始两个玩家轮流操作，每一次操作可以沿着边移动到一个未被到达过的点，无法移动的人会输。两个人都以最优策略移动，问有多少个出发点对第一个人来说是必胜的。 T是数据组数，$T \le 3; n \le 2000; m \le 3*10^5$.		一个点是必胜的当且仅当存在一个原图的最大匹配使得这个点不在最大匹配中。于是我们可以先用带花树求一遍最大匹配。可以发现和当前任意孤立点存在一条长度为偶数的交错路的点都可以变成孤立点，于是我们从每个孤立点出发再做一次增广，把所有距离为偶数或者花中的节点都标记为先手的人必胜的点。最后直接统计点数即可。	
时间复杂度	$O(Tnm)$	空间复杂度	$O(m)$

Problem 89

试题名称	Future of draughts		
试题来源	Codechef AUG 15	试题编号	CLOWAY
题目大意		算法讨论	
给定T张n_i个点m_i条边的无向图，有Q组询问，每组询问只考虑编号在$[L_i, R_i]$内的图，对于每张图选择一个出发点，接下来每回合选中一些图(至少一个)，并对这些图中的选中的点沿着边移动一步。问在K_i回合内每张图都回到出发点的方案数。模10^9+7。 $n, T \le 50; K_i \le 10^4; Q \le 10^5$.		先处理每张图长度为k的回路个数。显然就是每张图邻接矩阵G_i的k次方的主对角线元素和。我们可以先对邻接矩阵求特征多项式即$f(x) = det(xI - G)$，$f(x)$的多项式写法可以通过插值或高斯消元获得。由于$f(G) = 0$，可以求出路径条数的线性递推式。这样就可以预处理了。 然后对于每一组询问，令$w_{i,j}$表示第i张图长度为j的回路个数，通过容斥可以得到恰好K回合内结束的方案数是$\sum_{i=1}^{K}\prod_{j=L}^{R}(-1)^{K-i} * w_{j,i} * \binom{K}{i}$，答案就是这个的前缀和，我们可以枚举$L, R$然后用任意模数NTT预处理答案，每次输出即可。任意模数NTT可以选取三个NTT模数然后通过CRT求得答案。	
时间复杂度	$O(Tn^4 + TnK + T^2K\log K)$	空间复杂度	$O(Tn^2 + K)$

Problem 90

试题名称	Simple Queries		
试题来源	Codechef AUG 15	试题编号	DISTNUM
题目大意		算法讨论	
给定一个长度为n的数列和m个操作： 1.定义S为区间$[L, R]$内出现的数字集合，求$\sum_{1 \le i < j < k \le \|S\|} S_i, S_j, S_k \mod 10^9+7$。 2.插入一个数。 3.删除一个数。 4.修改一个位置的值。 5.询问一个区间内出现过的不同的数字个数。 $n, m \le 10^5$.		我们先用平衡树离线维护一下数字的相对顺序，这样就把2,3操作转化成了4操作。 我们对于每个元素A_i我们维护一个pre_i表示上一个出现的相同元素的位置，如果没有则$pre_i = 0$，接着我们把每个元素看成平面上的点(i, pre_i)，则一个询问$[L, R]$如果只要不同的元素就是满足$x \in [L, R] \& y \in [0, L-1]$的点的集合。我们采用树状数组套动态开节点的线段树维护。 至于操作1的询问内容，我们可以维护权值和S_1，平方和S_2，立方和S_3，则答案就是$\frac{S_1^3 - 3S_1S_2 + 2S_3}{6}$，而$S_1, S_2, S_3$都是很好合并的。	
时间复杂度	$O(m \log^2 n)$	空间复杂度	$O(m \log^2 n)$

以上是我90道传统试题的解答

以下是Challenge类型试题的做法

Challenge Problem 1

试题名称	The Great Plain		
试题来源	Codechef OCT 11	试题编号	LAND
题目大意		算法讨论	
给一个$n*m$的部分元素被确定的矩阵，你需要填上剩下的元素，矩阵中每个元素的范围是$[1,50]$。定义一个矩阵的权值为每对四联通相邻元素的贡献的和，两个元素a,b的贡献为$2^{abs(a-b)}$。 $10 \le n,m \le 100$.		假设我们已经有了一组初始解，考虑优化这个解。 我们扫描矩阵的所有元素，每个元素根据它周围四个元素可以在$[1,50]$内选取一个最优的值。这样反复迭代若干次就可以得到一个稳定的局部最优解。 但是这样的迭代可能会陷入比较劣的”死锁“中，我们可以考虑随机，每次迭代之前随机$\frac{nm}{60}$个元素将它们减去1。迭代次数可以选取$C=\frac{2*10^6}{nm}$，注意最后20次迭代之前不用随机化。 接着我们可以发现，一个元素跟棋盘上的确定元素之间产生的权值与它跟非确定元素之间产生的权值权重是不一样的，非确定元素可以改而确定元素不行，我们可以考虑乘上一个权值例如$6:7$。这个比例可以根据$n*m$的大小选取，$n*m$越大这个权值的差别就越大，当$n*m$很大的时候可以设为$2:3$。 初始解可以每个未确定格子选取最近的已确定格子，或者全部去某个值，影响并不大。	
时间复杂度	$O(TCnm)$	空间复杂度	$O(nm)$

Challenge Problem 2

试题名称	Stepping Average		
试题来源	Codechef NOV 11	试题编号	STEPAVG
题目大意		算法讨论	
给n个数A_i和一个整数K每次选择两个数合并成它们的平均数，操作$n-1$次后得到一个数，你要使得这个数尽可能的接近K。 T为数据组数，$T = 10; n = 1000; 1 \le A_i, K \le 2^{30}-1$.		每次求最大值R和最小值L，令$M = \frac{L+R}{2}$，设定阈值$W \in [0, 0.5]$，令$ML = L*(1-W)+R*W$，$MR = L*W+R*(1-W)$。 如果$K < ML$，讲L作为最后操作的数，目标转化把除了L以外的所有数合成的接近$2K-L$。 如果$K > MR$，讲R作为最后操作的数，目标转化把除了R以外的所有数合成的接近$2K-R$。 如果$ML \le K \le ML$，就直接把L, R合并。 这样问题的规模每次就减少了1，当问题规模缩减至1的时候就解决了。 W的取值可以调参，我的程序取了$W = 0.4$。	
时间复杂度	$O(Tn^2)$	空间复杂度	$O(n)$

Challenge Problem 3

试题名称	Similar Graphs		
试题来源	Codechef APRIL 12	试题编号	SIMGRAPH
题目大意		算法讨论	
给两张点数为n的图，你需要求一个点集的映射，使得映射过之后边集的交集尽可能大。 T为数据组数，$T = 20; 30 \le n \le 75$.		我们直接考虑使用模拟退火，每次我们尝试交换两个顶点，然后看答案是否更优，更优就接受，否则就以e的概率接受，e刚开始为1每次交换之后都乘上$w = 0.997$。退火$K = 4500000/n$次就可以得到局部最优解。 我们再把退火过程重复$C = 6$次取最优值即可。	
时间复杂度	$O(TCKn)$	空间复杂度	$O(n^2)$

Challenge Problem 4

<table>
<tr><td>试题名称</td><td colspan="3">Closest Points</td></tr>
<tr><td>试题来源</td><td>Codechef JUNE 12</td><td>试题编号</td><td>CLOSEST</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给定一个大小为N的三维空间的点集，有Q个询问，每次询问一个点的欧几里得最近点，按照正确率给分。
$N, Q \le 50000; |x|, |y|, |z| \le 10^9$.</td><td colspan="2">这是经典问题，我们可以直接使用KD树，每次划分的时候可以选择方差最大的那一维进行划分。为了防止TLE，可以强制每次询问只在KD树中找200个点就跳出。
另外要注意欧几里得距离的平方是会超过long long范围的。</td></tr>
<tr><td>时间复杂度</td><td>$O(n \log n + Qn^{\frac{2}{3}})$</td><td>空间复杂度</td><td>$O(n)$</td></tr>
</table>

Challenge Problem 5

<table>
<tr><td>试题名称</td><td colspan="3">Simultaneous Nim</td></tr>
<tr><td>试题来源</td><td>Codechef SEPT 12</td><td>试题编号</td><td>SIMNIM</td></tr>
<tr><td colspan="2">题目大意</td><td colspan="2">算法讨论</td></tr>
<tr><td colspan="2">给一个大小为n，异或和为0的多重集A，你需要把A划分成尽量多的异或和为0的若干个集合。
T为数据组数，$T = 10; 10 \le n \le 1000; 0 \le A_i \le 2^m - 1; 5 \le m \le 60$.</td><td colspan="2">首先我们可以划分出大小为2或者3的子集，分别使用暴力和map即可。
之后我们每次把未划分的元素随机一个顺序，依次添加元素，使用高斯消元维护异或方程组，直到方程组有解。
大约m个元素就能找到一个有解的方程组，这样就能把这个解求出来，把当前集合划分成两个，然后两个集合递归做下去，直到不能继续划分为止。
集合可以采用bitset来表示。</td></tr>
<tr><td>时间复杂度</td><td>$O(Tnm)$</td><td>空间复杂度</td><td>$O(n^2)$</td></tr>
</table>

Challenge Problem 6

试题名称	Maximum Sub-rectangle in Matrix		
试题来源	Codechef OCT 12	试题编号	MAXRECT
题目大意		算法讨论	
给一个$n*m$的矩阵A，求权值尽量大的子矩阵。 子矩阵的定义是一个行的子集和一个列的子集的笛卡尔积。 $200 \le n, m \le 300; \|A_{i,j}\| \le 10^9$.		假设我们已经有了一个初始解，我们可以采用迭代优化。先固定行集合，求出在当前行集合下哪些列的权值为正，更新列集合。再固定列集合，求出哪些行的权值为正，更新行集合。可以证明迭代在有限步数内能够结束，事实上迭代次数C很小，跑的很快。 初始解我们可以随机，一个元素被加入的概率为矩阵中大于0的元素个数与nm的比值。求出当前最优解之后采用当前最优解的$R_{size}C_{size}$与nm的比值。 随机K次去最优解即可。	
时间复杂度	$O(KCnm)$	空间复杂度	$O(nm)$

Challenge Problem 7

试题名称	To challenge or not		
试题来源	Codechef JUNE 13	试题编号	CHAORNOT
题目大意		算法讨论	
给一个大小为n集合A，求尽量大的子集使得不存在三个元素呈等差数列。 $1 \le n, A_i \le 10^5$.		首先尝试贪心做法，将A升序排列之后依次添加，用一个bool数组维护每个数能否添加，每加入一个数x，我们就扫描已经在答案集合中的数c，把$2t-c$设为不可加入。这样复杂度是$O(n+K^2)$的，可以发现K不是很大，只有10^3级别，所以跑的很快。这个解已经很优了，考虑加入随机化。如果一个数可以加进去，我们有p的概率不把它加入，这样有概率得到更优的答案。我们可以重复多次取最优解。 我的程序取了$p = \frac{1}{70}$，并且重复了700次取最优解。如果想继续优化，还可以对不同的n调一调参数，取不同的p即可。	
时间复杂度	$O(700(n+K^2))$	空间复杂度	$O(n)$

Challenge Problem 8

试题名称	Deleting numbers		
试题来源	Codechef AUG 13	试题编号	DELNMS
题目大意		算法讨论	
假设当前有n个数形成的数组a_i，每次可以选择v,t两个数，满足$v+t<=n+1$，且令k为最大的满足$v+kt<=n$的数，必须满足$a_v=a_{v+t}=a_{v+2t}=\cdots=a_{v+kt}$。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$n-k$个数形成的长度为$n-k$数组a_i，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。使用尽可能少的步数删除所有元素。		如果预先考虑删除之后下标变化处理起来十分麻烦，直接从尾部开始删。 每次删除从末尾元素t开始往前找尽可能长的由t组成的等差数列，把这个等差数列删掉，一直把数组删空。 这个解用来解决一般的随机数据已经很优秀了，不过可以考虑继续优化。 我们可以考虑末尾S个元素，如果其中出现次数最多的元素超过了B，我们用之前的方法把其他元素都删掉之后把这种元素一次性删掉。S,B的取值可以通过调参获得。 下标可以通过树状数组维护，i的下标为i减去i之前被删除的元素个数。	
时间复杂度	$O(n\log n)$	空间复杂度	$O(n)$

Challenge Problem 9

试题名称	Sereja and Permutation		
试题来源	Codechef APRIL 14	试题编号	SEAPERM
题目大意		算法讨论	
给定一个长度为n的数组A'，正整数K和整数S，你需要把数组重新排列为A。 令函数$f(A,i)=S-\sum_{k=i}^{j} A_k$，其中$j$为使得$\sum_{k=i}^{j} A_k \le S$的最大整数，若$A_i > S$则$f(A,i)=S$。 请你最小化$\frac{\sum_{i=1}^{K} f(A,i)}{K}$。 T为数据组数，$T \le 10; n \le 2000; A'_i \le 10^4$.		如果我们已经确定了第$K+1$个数到第n个数，我们可以贪心的选取前K个数，方法如下： 我们令k从K到1，依次最小化$f(A,k)$，具体的我们可以维护一个右指针r，初始值$r=n$。每次我们找一个最大的还没有被选取的数w满足$w+\sum_{j=k+1}^{r} A_j \le S$，如果存在合法的$w$我们就把$r$减小1，当$r=i$时说明剩下的数都超过$S$，这时随意放置都不会有区别。 至于第$K+1$个数到第$n$个数如何确定，我们可以首先去最小的若干个数升序排列作为初始解，然后随机$C/2$次，每次随机选取后$n-K$个数，利用贪心求出序列，更新答案。之后我们再随机$C/2$次，每次取当前最优解，对于后$n-K$个数，我们去p的概率把它跟前K 个数互换，这样就能不断优化当前最优解，得到一个非常优秀的答案。 如果采用平衡树寻找w，复杂度是$O(Cn\log n)$的。	
时间复杂度	$O(TCn\log n)$	空间复杂度	$O(n)$

Challenge Problem 10

试题名称	Kali and Devtas		
试题来源	Codechef DEC 14	试题编号	KALKI
题目大意		算法讨论	
给定大小为n的平面点集，求权值尽量小的生成树，权值定义为$\sum C_i$ C_i的定义是：对于每个点，设在生成树中与其相邻的点钟最远的点的距离为R，那么以该点为圆心，半径R以内的的点的Ci全部增加1（包括自身）。 $T \le 100; n \le 400.$		直接求平面最小生成树已经是非常优秀的解了，足以通过Tsinsen上的数据。 考虑优化，我们可以让每个点的度数尽可能的小，所以在取最小边的时候可以把连接的两个点的度数乘以相应的参数加到边权里去。选取边的时候也可以引入适当的随机化再重复多次取最优解。	
时间复杂度	$O(Tn^2\log n)$	空间复杂度	$O(n^2)$

以上是我90道传统试题+10道Challenge试题的泛做表格。