|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **题号** | **题目名称** | **题目大意** | **算法讨论** | **复杂度** |
| KNGHTMOV | KNGHTMOV骑士移动 | 有两种步长(Ax,Ay)和(Bx,By)的跳马问题，一些格子是障碍，问(0,0)到(X,Y)的移动方案数。 | 若两步长无关，则转换坐标后按x+y排序并DP。若两步长相关，则直接在一条线上DP解决。 | 时间O(N^2)，N是坐标范围  空间O(N) |
| PARADE | PARADE年度游行 | 在一张有向图上选择若干条有向路径，每条路径花费边权和，非环路径和未被路径经过的点均花费C，给定K个C求最小花费。 | floyd建最短路径图，以保证选出路径不相交。仿照路径覆盖问题用费用流建模，每个流量都会减少C，加上该流量自身费用。逐个增加流量，在询问时二分求解。 | 时间O(N^3 + maxflow + KlogN)  空间O(N^2) |
| SIMNIM | SIMNIM多重NIM游戏 | 给定N个xor和为零的数，尽可能多地将其拆分成xor和为零的子集。 | 每次用高斯消元法解异或方程组，求出一个xor为零的子集。随机化以优化答案。 | 时间O(NM^2)，M为二进制位数  空间O(N+M^2) |
| TWOCOMP | Two Companies | 给定一棵N个点的树，A,B两公司分别在树上拥有M1,M2条路径。选出一些路径，使得两公司的路径不相交，且权值和最大。 | 用LCA判断两路径是否相交。转换成最小割模型，A公司路径在S集代表选中，B公司路径在T集代表选中，用容量无限的边保证选中路径不相交。 | 时间O(NlogN + M1\*M2 + maxflow)  空间O(NlogN + M1\*M2) |
| SEAARC | Sereja and Arcs | 一条线上N个点，每个点有颜色。同色点间在直线上方连一弧线。求异色弧线交点个数。 | 分块，“大交大”枚举两个颜色，扫描位置。“大交小”枚举大颜色和小颜色的位置。“小交小”一遍扫描，枚举每个颜色的两个位置。分块阈值设为400. | 时间O(Nsqrt(N))  空间O(N) |
| BICKER | Bickering Cooks |  |  |  |
| AMBIDEX | [Ambidextrous Chefs](http://www.tsinsen.com/D11242) |  |  |  |
| CARDSHUF | [Card Shuffle](http://www.tsinsen.com/D11243) | 一堆牌，初始次序1~N。进行M次洗牌，每次将中间若干张抽出，倒序后插入牌堆中间的另一个位置。给出每次洗牌参数，求最终牌堆。 | 裸的平衡树维护序列。splay即可。 | 时间O(N + MlogN)  空间O(N) |
| MISINT2 | Misinterpretation 2 | 对一个字符串规定变换：将偶数位置上的字符提到前面，比如abcde变换后成为bdace。求长度[L,R]之间，变换后不动的小写英文字符串个数。答案模1e9+7. L<= R <= 10^10 , D=R-L <= 50000. | 变换视作排列，设长度n的循环数为g(n)。经数学分析，g(2n) = g(2n+1)-1，g(2n+1) = ∑phi(d) / ord2(d)，遍历d|2n+1，ord2是2关于d的阶。d用DFS遍历，过程中维护phi和ord2，为此需要求得ord2(p^k)，p是质数。将p-1质因数分解，逐个因子试探得t=ord2(p-1)，而ord2(p^k)=t\*p^b，b>=0.小质数预处理中直接分解p-1，在遍历素数表分解[L,R]时，将范围扩为[L-sqrt(R),R]，对于大质数p|d这一范围内必有p-1的倍数。 | 时间O((sqrt(R)+D)\*loglog(sqrt(R)) +sqrt(M))  空间O(DlogD+sqrt(R)\*log(sqrt(R))) |
| SEAEQ | Sereja and Equality | 我们称两个长度相等的数组相似，如果各元素的大小关系相同。对长度N的排列P1,P2，定义f(P1,P2)为所有使得P1[l~r]和P2[l~r]相似，且其中逆序对数不超过E的(l,r)数量。给定N<=500,E<=10^6，求P1,P2取遍所有排列的f(P1,P2)之和。有T<=10000组数据。 | 定义dp[n][m]为长度n，含m个逆序对的排列数。设dp[n][0~m]之和为dps[n][m]，那么dp[n][m]=dps[n-1][m]-dps[n-1][m-n]，据此可以一遍O(n^3)（注意m<=n^2）计算出所有的dp值，可以用滚动数组。然后对于每组数据，枚举l~r的长度，利用相应的dps可以很容易计算出有多少组(P1,P2)拥有相似的l~r，加在一起就是答案。 | 时间：O(N^3)  空间：O(TN+M) |
| GNUM | Game of Numbers | 给定两个长度N<=400的数组A,B。有两个数对集合S1,S2初始为空。每次选择两个数对(i,j)∉S1,(p,q)∉S2，使得Ai<Bj,Ap>Bq，四个数的gcd>1。然后将(i,j)加入S1，(p,q)加入S2.求最多操作次数。 | 网络流模型。将所有gcd(Ai,Aj)>1的(i,j)均作为节点，若Ai<Bj就放在二分图左边，Ap>Bq就放在右边。S向左边点连容量1的边，右边点向T连容量1的边。左边点向其gcd的所有素因子连容量1的边，右边点类似只是方向相反。最大流就是答案。需要卡常数：合并所有gcd相同的节点。 | 时间：O(maxflow(N^2,N^2logN))  空间：O(N^2logN) |
| GERALD09 | [Chef and Rectangle Genome](http://www.tsinsen.com/D11247) |  |  |  |
| EFFPAINT | Efficient Painting |  |  |  |
| QUERY | Observing the Tree | 给一棵点带权树（初始全0），有三种操作：①X~Y路径上，第一个点加A，第二个点加A+B，第三个点加A+2B……②查询X~Y路径上点权和。③回滚到某个操作之后的状态。强制在线，N,Q<=10^5. | 树链剖分，用类似“每个点深度\*B”的方法解决B的问题。为处理回滚操作，将所有剖分链顺次写在一起形成一棵大线段树（和NOI2015“软件包管理器”所用技术相同），并可持久化之。可持久化线段树不能进行lazy标记下传和子树信息合并（push\_down/up），区间加可用查询时携带lazy信息的方法解决。 | 时间O(Q\*log^2N)，空间O(Q\*log^2N) |
| ROC | Room Corner |  |  |  |
| FIBTREE | Fibonacci Numbers on Tree | 给一棵点带权树（初始全0），有四种操作：①x~y路径上第一个点加F[1]，第二个点加F[2]……F是斐波那契数列。②视x为根，求y子树的权值和。③求x~y路径权值和。④恢复到某次操作后的状态。强制在线，N,Q<=10^5 | 树链剖分。和NOI2015“软件包管理器”类似，将所有剖分链写在一起形成一棵大线段树，则某点子树在其上必为一整段。对于操作①，只需要记录某段第一、二个点（实际操作中是第一个点和它前面那个‘点’）加的数，即可用预处理结果迅速计算区间和/某点值。将线段树可持久化，即可执行所有操作。 | 时间O(Qlog^2N)  空间O(Qlog^2N) |
| QRECT | Rectangle Query |  |  |  |
| TILT | Table Tilt |  |  |  |
| CNTHEX | Counting Hexagons |  |  |  |
| SHORT | Short |  |  |  |
| EDSTGRID | Edit Distance on Grid |  |  |  |
| DEG3MAXT | Three-Degree-Bounded Maximum Cost Subtree |  |  |  |
| FN | Fibonacci Number | 给出模10余1或9的奇素数P和非负整数C，求最小的n使得Fib[n]=C (mod P),Fib是斐波那契数列 | 考虑斐波那契数列的通项公式。用Tonelli-Shanks“开根号”，从而求根号5，根据通项公式列出方程，用该算法解一元二次方程，再用大步小步算法求离散对数，最终求解。 | 时间O(sqrt(P))  空间O(hash) |
| SHOPPING | Shopping |  |  |  |
| CBAL | Chef and Balanced Strings |  |  |  |
| GRAPHCNT | Counting on a directed graph | 给一张有向图。求有多少个点对(x,y)，使得存在1~x和1~y的两条不相交（除了1）路径。N<=10^5,M<=5\*10^5. | 用Lengauer and Tarjan算法求控制树。控制树上LCA为1的两点均为合法点对。故只需求出1的每一棵子树大小即可。 | 时间O((N+E)logN)，空间O(N+M) |
| SEASORT2 | Sereja and Sorting 2 |  |  |  |
| STREETTA | The Street | 有1~N号商店，三种操作：①l~r号商店，i号加入一个价格为c\*i+d的商品。②l~r商店，i号消费税增加c\*i+d。③询问x号商店最贵商品加上消费税的值。N<=10^9，操作次数Q<=3\*10^5. | “商品”和“消费税”各自独立，所以这是两道题。开一棵线段树，记录该段的c,d，即可处理消费税。至于商品，也是用这样的线段树，在插入时，若新商品全部优/劣于该段的lazy标记，则酌情更新或放弃。否则，将它们看做直线，必然至多有一个交点。将lazy或当前商品作为新的lazy，另一个向交点所在一侧递归插入。N非常大，所以要么离散化，要么把线段树做成动态开点。 | 时间O(QlogN)，空间O(QlogN) |
| GERALD07 | Chef and Graph Queries | 有N个点，M条边，编号1~M。有Q个询问，每次询问只加入l~r的边后，图中联通块数量。N,M,Q<=2\*10^5. | 类似分块莫队算法，将边分成sqrt(M)块，询问先按左端点所在块，再按右端点排序。一次处理左端点在同一块的询问。这样右端点只会递增，但左端点不一定，所以每次需要撤销左端点变动造成的所有修改，将并查集可持久化即可实现这一点，方法是记录并查集数组的每一次变动，撤销时倒着改回去即可。由于左端点的变动范围不大，空间需求很小。需要注意左右端点在同一块的情况，但这无伤大雅。 | 时间O((Q+M)sqrt(M))（视并查集操作为常数），空间O(N+M+Q) |
| TAPAIR | Counting the Important Pairs | 给一张连通的简单无向图（无重边自环），问有多少种方式，删去两条边，使得这张图不再连通。N<=10^5,M<=3\*10^5. | 这道题和URAL 1557“网络攻击”是几乎一样的，但数据范围比后者大，而且多了“简单图”这个条件。进行一次DFS，由于是无向图，所以只有树边或回边，而没有横叉边。删去的两条边有三种更可能：①一条桥和任意一条边，②一条树边和覆盖它的唯一回边，③两条被同样一组回覆盖的树边。对每条回边，我们都随机一个在long long范围内的权值，然后使用类似“前缀和”的做法，我们可以求出每条树边被多少条回边覆盖，以及这些回边的权值异或起来是多少。利用前者，可以求出①和②（桥就是被0条回边覆盖的树边），利用后者可以求出③（若两条树边的异或和相等，就认为它们被同一组回边覆盖），于是问题就解决了。这种巧妙的哈希颇为有趣。 | 时间O(N+M)，空间O(N+M). |
| CNTDSETS | Counting D-sets | 给定N、D，求N维空间中有多少个直径为D的不同整点集。直径定义为点集中两点距离的最大值，此处距离指切比雪夫距离，即各坐标之差的最大绝对值。若两个点集可在平移后重合，则认为它们相同。N<=1000,D<=10^9. | 显然只需要求“直径<=D的点集数”，因为把D的结果减去D-1的结果就是我们想要的答案了。至于“平移重合”的问题，我们只需要简单地规定①所有点的坐标在[0,D]内，②对于每一维，都有某个点的这一维坐标是0。可以发现满足①②的两个点集不会在平移后重合，并且也必然满足直径<=D这个条件。怎么保持②呢？容斥。对i=0~N，可以轻松计算已知有i维不符合②（即没有任何点的这一维坐标是0）的点集数量S(i)：C(N,i)\*2^(D^i\*(D+1)^N-i).其中，D^i\*(D+1)^N-i就是有i维的坐标范围是1~D，另外N-i维的坐标范围是0~D的高维长方体包含的点数。那么答案自然就是S(0)-S(1)+S(2)…了。D^i这样的值可能很大，注意到MOD是一个素数，所以先做一次快速幂，算出它模MOD-1的值，然后再作以2为底的快速幂。 | 时间O(N^2+Nlog(MOD))，空间O(N^2) |
| TOURBUS | Byteland Tours |  |  |  |
| CHECKERS | Eastern Draughts |  |  |  |
| GTHRONES | A Game of Thrones |  |  |  |
| MAGIC | Two Magicians | 有一个N<=7777个点，M<=10000条边的简单无向连通图。甲乙两人在上面游戏。初始时分别在1,2，二人均由P<=10000个魔法点数，甲先手。每回合进行三个操作：①走到所在连通块内任意一点（若走到另一人所在位置，胜）。②连接一条图中没有的边。③（可选）消耗一个魔法点数，跳到途中任意一点。问谁胜。 | 博弈题……有规律。设E=N\*(N-1)/2-M（也就是添几条边能成完全图）。设大小为偶数的连通块数量为X，大小为奇数的连通块数量为Y。反正规律就是这几个数来回判断…… | 时间O(N\*α(N))，空间O(N) |
| CLOSEST | Closest Points |  |  |  |
| MATCH | Expected Maximum Matching |  |  |  |
| COOLNUM | Cool Numbers |  |  |  |
| TOMJERRY | Tom And Jerry |  |  |  |
| CIELQUAK | Ciel and Earthquake |  |  |  |
| EVILBOOK | Evil Book |  |  |  |
| DGCD | Dynamic GCD |  |  |  |
| ADDCHAIN | Addition chains |  |  |  |
| EST | Equivalent Suffix Tries |  |  |  |
| MINESREV | Minesweeper Reversed |  |  |  |
| CHANGE | Making Change |  |  |  |
| LECOINS | Little Elephant and Colored Coins |  |  |  |
| HAMILG | A game on a graph |  |  |  |
| MAXDIFFW | Maximum Difference Walk |  |  |  |
| EASYEX | Easy Exam | 有一个1~K的均匀骰子，掷N次，设ai为掷出i的次数，求(a1^F)\*(a2^F)\*…\*(aL^F)的期望值。答案模2003（除法就是乘以乘法逆元），N,K<=10^9,L\*F<=20000,F<=1000. | 将ai分解成N个值的和：xi,1+xi,2+…+xi,N，其中xi,j是一个01数，代表第i次掷出的是否为j。于是问题转化为E[(x1,1+x1,2+...x1,N)F∗(x2,1+x2,2+...x2,N)F∗...∗(xL,1+xL,2+...xL,N)F].展开后，结果中每一项所含的不同未知数的个数是L~L\*F，可以枚举之。为此需要知道，对于某个P，总共有多少项含有P个不同未知数。可以用生成函数求得，具体来讲，就是一个以斯特林数为系数的多项式的L次方，其x^P的系数就是要求的值。该多项式的次数就是P的最大值，即L\*F，因此可以使用FFT+快速幂解决。但注意到模数很小，故对于P>=2003，它对最终答案的贡献一定是0，所以直接舍弃掉这些项，直接暴力做多项式乘法即可。 | 时间O(MOD^2\*logL+F^2)，空间O(MOD +F^2). |
| DELNMS | Deleting numbers |  |  |  |
| LYRC | Music & Lyrics | 给W<=500个长度P<=1000的模式串，再给N<=100个长度S<=50000的文本串，问每个模式串在每个文本串中出现次数的总和。 | 先把所有模式串建成AC自动机，那么就有一棵fail树。然后把所有文本串放上去匹配，记录下AC自动机每个节点被访问的次数cnt。那么，假设某个模式串的末尾节点是pos，该模式串的出现次数总和就是pos在fail树上子树所有节点的cnt之和。所以按BFS序倒着累加一下即可（利用建fail指针时的BFS序）。 | O(输入规模)，即O(WP+NS) |
| PRIMEDST | Prime Distance On Tree |  |  |  |
| CHAORNOT | To challenge or not |  |  |  |
| SPMATRIX | Count Special Matrices | 求满足下列条件的N\*N矩阵个数：  1.对称，主对角线为0，其余元素非零  2.A[x][y]<=max(A[x][z],A[y][z])对所有1<=x,y,z<=N成立  3.不在主对角线上的元素取1~N-2，且1~N-2均出现过  答案模10^9+7，N<=10^7. | 首先暴力求出来几项，然后可以发现OEIS上有这个数列……通项是N!(N-1)!/(2^(N-1))\*(N/2-2\*3-H[N-1]/3)，其中H是算术级数，H[k]=1+1/2+…+1/k.因此预处理阶乘，分块预处理(1/2)^i，就可以解决问题了。通项有一个很复杂的证明，需要用到一些抽象代数知识。 | 预处理O(N)，单次查询O(1)，空间O(N) |
| TKCONVEX | Two k-Convex Polygons | 给定N<=1000个长度是<=10^9的整数的棍子。问能否选出两组，每组K<=10个，每组摆成一个凸多边形。 | 首先注意到，K根棍子能摆成凸多边形的充要条件是：这K根中最长棍子的长度严格小于其余的长度之和。由此得到一个推论：如果将棍子按长度从小到大排序，那么这2K根棍子，要么是分开的两组，每一组是连续的K根，要么就是连续的2K根，其中K根属于第一组，K根属于第二组。还有一个推论：只要有O(log(10^9))根棍子，就一定能选出呈凸多边形的K根（想象一下，‘恰好不能形成三角形’的是Fibonacci数列，呈指数增长）。所以我们先扫描判断是否存在“分开的两组”这种（排序后一遍扫过去就可以知道每组连续K根是否可行），若不行，就枚举所有连续的2K根，暴搜是否可行。 | 时间O(log(10^9)\*C(2K,K)+NlogN)，空间O(N) |
| STEPAVG | Stepping Average |  |  |  |
| LUCKYDAY | Lucky Days |  |  |  |
| DOMNOCUT | Colored Domino Tilings and Cuts |  |  |  |
| LAND | The Great Plain |  |  |  |
| PARSIN | Sine Partition Function |  |  |  |
| BAKE | The Baking Business |  |  |  |
| STRQUERY | String Query |  |  |  |
| INVBINCF | Inverse Binomial Coefficient |  |  |  |
| FAULT | Fault Tolerance |  |  |  |
| MAXCIR | Max Circumference |  |  |  |
| CIELHACK | Ciel and password cracking |  |  |  |
| MAXRECT | Maximum Sub-rectangle in Matrix |  |  |  |
| RANKA | Ranka | 给一个正整数N，求一盘步数为N的围棋（9\*9）棋局走法。要求不能出现同形反复。执棋者相同且棋盘状态相同就叫同形反复。N<=10000. | 先用白子填满(1,1)之外的所有格，然后黑子填(1,1)，吃掉所有白子；然后用白子填满(1,1),(1,2)之外的所有格，然后黑子填(1,2)，吃掉所有白子……如此下去，直到黑子填满(9,9)外所有格。这一轮大概能走6000多步。然后用白子填(9,9)，吃掉所有黑子，黑白角色互换，再像刚才一样填一轮，一共就是12000多步，足够了。 | 时间O(N)，空间O(1) |
| XRQRS | Xor Queries | 对一个初始为空的序列，有四种操作：①在序列末端加入x，②在区间[L,R]内找出异或x最大的数，③删除序列最后x个元素，④在区间[L,R]内统计<=x的数字个数，⑤求区间[L,R]内的第x小值。操作数M和x都<=5\*10^5. | 用可持久化Trie。用两个代表某前缀的Trie节点相减，就是一段的Trie了。①是显然的。②直接从上往下，尽量走和x不一样的。③是显然的。④改为统计<x的个数，那么从上往下走，逐个加起来即可。⑤和普通的树上二分一样。此题清澄上卡常数，需快速读入。 | 时间O(MlogM)，空间O(MlogM) |
| SEAND2 | Sereja and Number Division 2 |  |  |  |
| CHEFPNT | Chef and Painting |  |  |  |
| BTREE | Union on Tree |  |  |  |
| TRIPS | Children Trips |  |  |  |
| GERALD08 | Chef and Tree Game | 给一棵N<=10^5的树，树边要么红要么蓝。有甲乙二人，轮流操作。每次甲删去一条红边，乙删去一条蓝边，删边后一并删去因此不再与根连通的所有点。无法操作者输。问甲先手和后手时谁胜。 | 这就是所谓的Blue-Red Hackenbush问题，对此类问题有一个类似SG函数的“状态值”，代表了谁能获胜。计算状态值的方法亦见诸相关文献。但这个值不一定是整数，它有可能是一个分母为2的幂的分数。因此需要用高精度来记录这些值，具体来说，就是用一个set存储二进制中所有1的位置。同时，为避免运算次数过多，在计算时应采用启发式的方法。 | 时间O(Nlog^2N),空间O(NlogN) |
| SEAPERM | Sereja and Permutation |  |  |  |
| LMATRIX3 | Make It Zero 3 |  |  |  |
| SEASNAKE | Sereja and Snake |  |  |  |
| TMP01 | To Queue or not to Queue |  |  |  |
| TWOROADS | Two Roads | 给定平面上N<=100个点。要求你画两条直线，使得所有点到最近直线距离的平方和最小。 | 如果只有一条直线，那么可以像最小二乘法的推导一样，设直线为y=kx+b，把式子写出来，先对b求偏导，以消去b，再对x求导，得出最终结果。两条直线怎么办呢？可以发现，它们的两个平分线一定互相垂直。因此相当于在平面上建立一个垂直坐标系，那么第一第三象限的点离一条直线更近，第二第四象限的点离另一条直线更近。进一步地，两条“坐标轴”一定满足：一个由点集中某两点确定，另一个过点集中某点（否则可以旋转平移它们直至符合条件）。于是可行的划分方案不超过N^3种。枚举一个“坐标轴”，将所有点按投影排序后再枚举另一个，就可以及时维护求最优值需要的诸多系数。 | 时间O(N^3logN)，空间O(N) |
| MARTARTS | Martial Arts |  |  |  |
| COUNTARI | Arithmetic Progressions | 给一个长度N<=10^5的数列，元素<=M=30000.问有多少种方法能从数列中选出三项，使得它们按顺序构成一个等差数列。 | 可以用分块+FFT做。不过也有一个神奇的做法：大暴力，就是枚举中间那个数，由于元素<=30000，直接维护等于每个值的元素有多少个，枚举等差数列的公差，从而计算答案。用指针怎么优化一下，就神奇的A掉了。 | 时间O(NM)，空间O(N+M). |
| MANYLEFT | Many Left |  |  |  |
| CLOWAY | Future of draughts |  |  |  |
| SCLUSTER | Social Cluster |  |  |  |
| DISTNUM | Simple Queries |  |  |  |
| ANUDTQ | Dynamic Trees and Queries |  |  |  |
| SEINC | Sereja and Subsegment Increasings | 给两个长度为N<=10^5的数组A,B，每次我们可以将A中连续的一段+1然后模4，问最少操作多少次能把A变成B。 | 令C[i]=(B[i]-A[i])%4（保证0<=C[i]<=3），则问题变成了：一开始数组是(0,0…,0)，每次可以将连续的一段+1，问最少操作多少次能变成C。和NOIP2013“积木大赛”一样，令C[0]=0, D[i]=C[i]-C[i-1] （1<=i<=N），答案就是D中所有的正数之和。但是！我们可以将C中的某一段+4，从而在不改变原问题（操作A是在模4意义下的，所以+4也一样）的同时减少操作次数。例如：C={3,0,3}时答案为6，但C={3,4,3}时答案为4，反而减少了。当然了，C中每个数也至多加一次4.每次“+4”操作，相当于选取1<=i<j<=N，将D[i]加4，D[j]减4（理论上j可以是N+1，但这不会让答案变小）。可以发现，会让答案变小的操作一定满足：D[i]<0且D[j]>0，这会让答案减小D[j]-D[i]-4，那么有意义的(i,j)一定使得D[j]-D[i]>4.换言之，我们在把D中的数进行配对，满足：①只有-2,-3,2,3可以参与配对，一个元素只能配对一次。②一定是前面一个负数和后面一个正数配对。③答案的减小值等于参与配对的-3和3的数量（将D[j]-D[i]-4写作(D[j]-2)+(-D[i]-2)就可以看出这一点）。因此使用贪心：先尽量将-3和3进行配对，一遍完成后，再尽量用-2和2分别和那些未配上对的3和-3配对。最终答案就是D中正数之和减去配上对的-3和3的数量。 | 时间O(N)，空间O(N) |
| ANUMFS | The Malaysian Flight Search |  |  |  |
| YALOP | Trial of Doom |  |  |  |
| BB | [Billboards](http://www.tsinsen.com/D11358) |  |  |  |
| KITCHEN | [Large Kitchen](http://www.tsinsen.com/D11453) |  |  |  |
| TICKETS | Selling Tickets |  |  |  |
| LEBOXES | Little Elephant and Boxes |  |  |  |
| CKROACH | Killing Gs |  |  |  |
| KILOWHAT | A Fence for Byteland |  |  |  |
| ANDOOR | A New Door |  |  |  |
| CUCUMBER | Cucumber Boy and Cucumber Girl | 给B个N\*N矩阵Q[1]~Q[B]。对a,b，定义N\*N矩阵Va,b，满足Va,b[i][j]=(Q[a][i]\*Q[b][j])%2。乘法符号代表两个行向量做点积。我们称无序对(a,b)是好的，如果N!-det(Va,b)是奇数。求好的对数。B<8000，N<=60. | N=1时特判一下就好。N>=2时，N!是奇数，所以只需要求使得det(Va,b)是奇数的(a,b)对数。定义大小为N\*(N+1)的01矩阵D[i]，其前N列是（模2意义下的）Q[i]，最后一列全是1.则Va,b=(D[a]^T)\*D[b].其中D[a]^T是D[a]的转置。应用Cauchy-Binet公式知，det(Va,b) = ∑det(V[a][i])\*det(V[b][i])，1<=i<=N.其中V[a][i]是D[a]除去第i列后，剩余N\*N矩阵的行列式，V[b][i]类似。怎么求V呢？对于一个N\*(N+1)的01矩阵A，对其做高斯-约当消元。如果秩小于N，则V全零。如果秩等于N，那么必有某个第p列无主元。若i<p，V[i]=A[i][p]（这里的A当然是消元完毕后的），若i=p，V[i]=1，若i>p，V[i]=0.如此便可求出V。注意到N<=60，我们全程用位运算加速。V[a]和V[b]都是long long型变量，那么det(Va,b)= \_\_builtin\_popcountll(V[a]&V[b]). 这是一个快速查询二进制位中有几个1的库函数。 | 时间O((N^3B+B^2N)/64)，空间O(N^2B/64). |
| COMPLEXT | Complex Spanning Tree |  |  |  |
| SHORTCIR | Shortest Circuit Evaluation |  |  |  |
| DIVISORS | Something About Divisors |  |  |  |
| CONPOIN | Connect Points |  |  |  |
| SEADIVM | Sereja and Matrix Division |  |  |  |
| CHEFBOOK | Chefbook | 给一张N<=100,M<=10000的有向图，边带权L(x,y)。有两个数组P,Q，代表将第i个点的出边权值加P[i]，入边权值减Q[i].对边(x,y)，要求操作后权值在[S(x,y),T(x,y)]之间。求操作之后所有边权之和的最大值，以及一组可行的P,Q。 | 不妨设P[N+i]=Q[i]，这样就只有一个向量P了。将该问题看做一个线性规划问题，约束形如S(x,y)<=L(x,y)+P[x]-Q[y]<=T(x,y)，要求最大化∑C[i]\*P[i]，其中对于i<=N,C[i]是点i的出度，对于N<i<=2N,C[i]是点i入度的相反数。将其写成矩阵的形式：约束是AP<=B,要求最大化T(C)\*P。T(C)是C（一个列向量）的转置。可以看出A一定是每行有且仅有一个1和一个-1.该线性规划有着对偶问题：约束是T(A)\*Y>=C,要求最小化T(B)\*Y.由于∑C[i]=0，可以发现必然有T(A)\*Y=C，并且T(A)每列有且仅有一个1和一个-1.可以用最小费用最大流解决这样的问题：每一行都用图中的一个点表示，每一列是一条边，从1的点指向-1的点，容量INF，费用为相应的B。如果C[i]>0，从S向i连容量C[i]费用0的边，否则从i向T连容量-C[i]费用0的边。可以发现此时矩阵乘法表示的每个等式就是一个点的流量平衡方程。那么求一遍最小费用最大流，若能流满，就证明有解，而且函数值等于总费用。怎么求答案呢？根据互补松弛定理，如果我们求出的Y[i]非零，那么原问题中的第i个约束一定取等号。再加上原先S(x,y)<=L(x,y)+P[x]-Q[y]<=T(x,y)这样的条件，用差分约束就可以解决了。可以将P数组整体加减一个值以保证都在[0,1e6]之间。 | 时间O(maxflow(N,M))，空间O(N+M). |
| FINDSEQ | Find a Subsequence |  |  |  |
| FLYDIST | Flight Distance |  |  |  |
| SMVSEVIL | Chef vs Evil Chef |  |  |  |
| SIGFIB | Team Sigma and Fibonacci | 给定N,M，求∑(6xyz\*F[x]\*F[y]\*F[z])%M，其中F是斐波那契数列，要求x+y+z=N。N<=10^18，M<=10^5，所有测试点中M之和<=10^6. | 这是一道非常数学的题。可以算出来答案（关于N）的生成函数，其系数满足一个12阶递推式，因此可以用类似快速幂的方法迅速求解。当然每次快速幂需要对数时间，因此可以先把一些N,M较小的解打表存起来，以免超时。 | 时间O(Testcase\*logN)，空间O(M) |
| PUSHFLOW | Push the Flow! |  |  |  |
| SEASHUF | Sereja and Shuffling |  |  |  |
| WORDNINJ | WordNinjas |  |  |  |
| QPOLYSUM | Quasi-Polynomial Sum |  |  |  |
| DIFTRIP | Different Trips | 有一棵N<=10^5个点的树。我们将某一点走到其某个祖先的路径称为“祖先路径”。我们称两条路径相同，当且仅当它们长度相同，并且对应点的度数相同。问有多少条不相同的祖先路径。 | 直接使用在树上的，广义后缀自动机即可。和ZJOI2015“诸神眷顾的幻想乡”类似，不过比后者简单。字符集大小O(N)，所以需要map存边。 | 时间O(NlogN)，空间O(N) |
| CHPUZZLE | Jigsaw Puzzle Solving |  |  |  |
| CUSTPRIM | Payton numbers | 定义三元组的乘法 　　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) 　　定义zero：若x\*任何y=0，则称x是zero 　　定义单位元，若x\*任何y=y，则称x是单位元 　　定义质数，若x不是zero且不能分解成两个非单位元的乘积，则称x是质数 　　给定一个三元组，问是不是质数 | 设ω=(1+sqrt(-11))/2，那么可以证明，如果把三元组(a,b,c)视作(33-2a-c)+(b-a) ω，那么这两种系统在运算上是同构的。因此只需要考虑在这种“x+ωy”系统下的“质数”。经过一系列数学推导，可以发现：若y≠0，那么x+ωy是质数当且仅当x^2+xy+3y^2是素数。若y=0，那么x+ωy是质数当且仅当①x=±2，或②|x|是奇素数且x≠±11，同时-11不是模x的二次剩余。因此用Rabin-Miller算法检测一个数是否为素数，用勒让德符号计算是否是二次剩余即可。可能涉及10^14量级的数相乘，因此对“取模乘法”需要特殊处理。 | 时间O(logN)，空间O(1) |
| DEVLOCK | Devu and Locks |  |  |  |
| QTREE6 | Query on a tree VI | 给一棵N个点的树，一开始所有点都为黑色。M个询问，格式为二者之一：①查询某个点所在的同色联通块大小。②改变某个点的颜色（黑变白/白变黑）。N,M<=10^5. | 树链剖分。每个点x定义white[x]和black[x]，代表若该点为白/黑色，在自己子树中的同色联通块大小。对于①，找到x的“最浅同色联通祖先”（这可以用线段树实现）u，然后答案就是u的white或black值。对于②，以x黑变白为例，找到上述u，将father[x]~father[u]这一段的black值减去black[x]，然后改变x的颜色，这时“最浅同色联通祖先”也会改变，设为v，将father[x]~father[v]这一段的white值加上white[x]。 | 时间O(Mlog^2N)，空间O(N) |
| REALSET | Petya and Sequence | 判断一个n\*n循环矩阵A是否满秩。n<=3\*10^4. | 根据数学结论，A不满秩，当且仅当A的关联多项式f(x)（associated polynomial，其实也就是A第一列的生成多项式）和x^n-1有超过零阶（即非常数）的公因式。借助分圆多项式进行一番推导后，发现A不满秩当且仅当存在某个d|n，使得x^d-1|f(x)\*Π(x^(d/p)-1)，其中p|d且p为素数。我们还发现，某个多项式模x^d-1，就等于把它所有的指数都模d的结果。这样直接算一下，若最后得出来系数全0就说明整除。 | 时间O(n\*logn\*d(n))，d(n)是n的因数个数。空间O(n). |
| SMPAINT | Art in Digital Age |  |  |  |
| TREECNT2 | Counting on a Tree | 给一棵n<=10^5的树，边带权值<=10^6。有q<=100个操作，每个操作是修改某条边的权值，之后问有多少对点(S,T)使得S~T路径上权值gcd为1. | 用莫比乌斯反演。按照d将图分层。10^6以内的数至多有c=7个不同质因子，因此至多涉及2^7层，总点数是O(128N)的。可以用并查集解决。先将所有未改过的边固定起来，然后暴力计算每个修改，算完之后把并查集倒回去（故需可持久化）。这样一共计算q^2次。 | 时间O(2^c\*(n+q^2)\* [α](https://zh.wikipedia.org/wiki/%CE%91)(n))，空间O(w+2^c\*n) |
| EMBED | Embedding |  |  |  |
| RNG | Random Number Generator |  |  |  |
| CAVE | Mushroom Cave |  |  |  |
| MINESREV | Minesweeper Reversed |  |  |  |
| CLONES | Attack of the Clones |  |  |  |
| SPELL | The Spelling Problem |  |  |  |
| SEAORD | Sereja and Order | 有N个任务，第i个需要在电脑A上运行A[i]秒，电脑B上运行B[i]秒。一台电脑同时只能运行一个任务，一个任务同时也只能在一台电脑上运行。求最少运行时间，及一组方案。N<=10000. | 首先有个结论：最少运行时间是max(max{A[i]+B[i]}, ∑A[i], ∑B[i]).若是max{A[i]+B[i]}，说明方案的“骨架”是某一个任务，先做A再做B，并把其他任务填充到空闲机时里。若是∑A[i]（∑B[i]是类似的），说明方案的骨架是所有任务的A机时以某种方式排列，一个接一个。于是可以随机化两台电脑上的任务顺序，让电脑A持续运行，并检测电脑B上的任务是否能在∑A[i]时间内结束（考虑到一个任务不能同时在两台电脑上运行的限制），多次随机直到找到一组可行解。 | 时间O(kN)，K是随机次数，空间O(N). |
| FNCS | Chef and Churu | 给一个数列A[1~N]，N个函数F[1~N]，F[i]返回A[L[i]~R[i]]之和。有Q个操作，两种：①修改某个A[i]。②求F[x~y]之和。N,Q<=10^5，A[i],y<=10^9. | 将F分成sqrt(N)块。预处理每一块之和，以及每个A[i]在每一块里出现了几次（此处需要用差分）。再用一个树状数组维护A的区间和。操作①：用预处理值修改每一块的和，以及树状数组。操作②：把整块的和直接加起来，零碎的在树状数组上单独计算每一个F值。答案需要用unsigned long long存。 | 时间O(N\*sqrt(N)+Q\*sqrt(N)\*logN)，空间O(N\*sqrt(N)). |
| SHORT2 | Short II |  |  |  |
| HYPER | Hypertrees | 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个带标号的点的本质不同的3-超树.N<=17. | N<=17，暴力打表……就行了。 | 时间O(N)，空间O(N) |
| SPIN | Spinning Wheels |  |  |  |
| BWGAME | Black-white Board Game | 给一个N\*N矩阵，第i行的L[i]~R[i]列为1，其余全是0.问该矩阵行列式的正负号。 | 从1到N遍历C，用可合并（小根）堆维护L[i]=C的所有行，取其中R最小者x（若不存在，或者有多个最小者，则行列式为零），消元。则除x之外，L[i]=C的其余L[i]都变成了R[x]+1（合并一下即可）。对该列做拉普拉斯展开，可以发现仅余子式M(C,x)是有意义的（这一列消元后就剩这一个1了）。因此就转化为去掉一行一列的问题了。用树状数组维护某一行当前的编号（因为我们要计算余子式），而当前列的编号一定是1。此题input文件甚大，需快速读入。 | 时间O(NlogN)，空间O(N). |
| DIVLAND | Division of Lands |  |  |  |
| LPARTY | Little Party | 有一个函数f(v)，输入是一个N维bool向量v，在其某M种取值下函数值为1，其余情况为0.我们试图找出f(v) ≡ A1(v)|A2(v)| … |Ap(v)，其中Ai(v)是形如“a&b&…&z”（每个变量是某一xk或者!xk）的函数，并使得∑len(Ai(v))（len，或‘长度’，就是这个函数包含的变量个数）最小。“|”是逻辑或，“&”是逻辑与。N<=5,M<=1000. | 首先明确一点：有效的M至多2^N而非1000，因为v一共只有2^N种取值。对于上述形如“a&b&…&z”的函数A(v)，若每个使其为1的v都使f(v)=1，则称A是f的一个“基元”。我们定义使A(v)=1的v集合是A的“真值集”，那么问题就变成了：找出若干个基元，使其真值集的并集等于f的真值集S（也就是输入的那M个），求这些基元长度之和的最小值。对于基元A,B，若A的真值集是B的真值集的子集，那么A一定不会出现在最优答案中，可以删去。如此操作后，有效的基元至多剩下32个（这是实验结果）。然后用暴搜+剪枝解决。剪枝如下：①按照长度从大到小地考虑各基元。②若当前长度之和已大于答案，退出。③若当前真值集并上后面所有基元的真值集不等于S，退出。④若加上当前基元并不会改变真值集，则不加。 | 时间O(4^N)，空间O(3^N). |
| CONNECT | Find a special connected block |  |  |  |
| TSUBSTR | Substrings on a Tree |  |  |  |
| SIMGRAPH | Similar Graphs |  |  |  |
| DAGCH | Graph Challenge | 给一张N<=10^5，M<=2\*10^5的有向图，使得从1开始按某种顺序DFS，可以让每个点的标号等于其DFS序号。求每个点的半支配点。 | 求支配树有一个Lengauer Tarjan算法，用其中求半支配点的部分即可。 | 时间O(MlogM+N\*α(N))，空间O(M+N). |
| COT5 | Count on a Treap |  |  |  |
| LMATRIX2 | Make It Zero 2 |  |  |  |
| QPOINT | Queries With Points |  |  |  |
| MONOPLOY | Gangsters of Treeland |  |  |  |
| SEAVEC | Sereja and Vectors |  |  |  |
| QTREE | Queries on tree again! |  |  |  |
| CPP | Chef Protection Plan |  |  |  |
| KALKI | Kali and Devtas |  |  |  |
| DIVIDEN | Divide or die | 给一个N°角，要求给出一种尺规作图步骤，将其N等分。 | 根据尺规作图的基本结论，我们可以做出36°角，也可以做出30°角，从而可以做出6°角，等分得3°角。所以如果N%3≠0，就可以用某个3的倍数和N作比较，从而得到1°角，问题解决。如果N%3=0，则不可做——因为这是三等分角问题。 | 时间O(N)，空间O(N). |
| RIN | Course Selection | 有N门课，M个学期。必须学完所有课程，每门课都需要学一学期。有K对依赖关系，每一对形如(a,b)，代表必须先学完a再去学b。若课程i在学期j学，则所得分数为0<=X[i][j]<=100。X[i][j]可以是-1，代表该学期无此课程。求最高总分。N,M,K<=100. | 最小割模型。将第i门课拆成M个点，顺次相连，首尾和S,T相连形成路径：S->v(i,1)->v(i,2)->...->v(i,M)->T。显然做完最大流之后，从S到某个v(i,j)的点归于S集，后面的归于T集。v(i,j-1)->v(i,j)的边（规定v(i,0)=S）容量是100-X[i][j]，割掉这条边代表在j学期学i，会从满分中扣掉多少。特别地，若X[i][j]=-1那么容量就是INF，代表不可能被割掉。这样建图完美符合了“每门课在且仅在一个学期学”的条件。推而广之，这是一种普适的“若干个中选取一个”的建图方法。依赖关系怎么办呢？v(a,0)->v(b,1), v(a,1)->v(b,2), ..., v(a,M-1)->v(b,M)，都连容量INF的边就解决了。然后做一遍最大流，得到的最小割值cut就是“至少扣掉多少分”，满分-cut就是最高总分了。 | 时间O(maxflow(NM,(N+K)M)，空间O((N+K)M) |