姓名：林锦坤

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试题编号 | 名称 | 题目大意 | 算法讨论 | 时空复杂度 |
| Usaco  Open  06 | Tower of Hay | 给定n (n<=100000)堆干草，每堆干草有一个长度。现在要用这些干草堆出一座塔，要求必须按顺序堆，并且上面的宽度不能超过下面的，求塔最高是多少。 | 将塔看作一个面积一定图形，要使其最高，必须最瘦。于是底边最短的方案，层数最高。F[i]代表用i到n的干草堆出的塔的底层最短是多少。F[i]=min(sum[i..j-1]) (f[j]<=sum[i..j-1])  用单调队列优化即可。 | O(N) |
| CF293D | Ksusha and Square | 给定一个n(n<=10^5)个点的凸包，坐标绝对值<=10^6。任选两个点，求以这两个点为对立顶点的正方形的期望面积。 | 其实就是要求下面这个式子：  可以x,y分开算。讨论x:  令ai代表x=i的点有几个，最后就是要求：  a2a1 + a3(a2 + 22a1) +  a4(a3 + 22a2 + 32a1) + ...  它们差是：  a2 + 3a1, a3 + 3a2 + 5a1,  a4 + 3a3 + 5a2 + 7a1  再次差分即可发现规律，按照规律递推即可 | O(N) |
| CF 306C | White, Black and White Again | 已知接下来n(<=4000)天会发生w件好事和b件坏事，每天只会发生一种类型的事，并且接下来n天只会先一直发生好事，然后坏事，然后好事。事件都是不同的。求有几种方案。 | 先假设事件之间是相同的，最后再乘上w!\*b!即可。  再算出把好事分成两组的方案数，然每种方案和坏事放在一起刚好分成三组，在三组的基础上算出分出n组的方案，最后再乘上 w!\*b!就是答案。 | O(N\*N) |
| CF 351D | Jeff and Removing Periods | 规定一个操作：对于一串数，你可以选择几个数值相同，并且位置的差分也要相同的数，然后将它们删除，之后任意排列数列。给定一个长为n的序列。q个询问，每次询问有两个数l,r代表在[l,r]这个区间上进行如上操作最少几次可以删光区间里面的数。 | 其实就是答案就是区间里面不同的数的个数-是否有一个数值它们位置的差分都相等。  然后离线处理，将询问按r排序，i从1开始扫到n。对于第一个问题，维护一个数组c1每次扫到一个数，就将pre[i]+1到i都加1。对于一个[l,i]的询问，第一个问题的答案就是c1[l]。第二个问题也用类似方法处理即可。 | O(NlogN) |
| CF 323B | Tournament graph | 你要构造一个有N个结点的竞赛图，使得对任意两个结点u和v（u≠v），从u到v的最短距离不超过2。 | 手动构造出n=3和n=6的解。对于一个n的解，可以构造出n+2的解。具体如下：1到n连向n+1，n+1连向n+2，n+2连向1到n。n=4无解 | O(N\*N) |
| USACO March Contest 2008, Gold Division, Problem 1 | Land Acquisition | FJ要买n块长方形土地，他可以分批购买。每一批需要支付maxh\*maxw的价钱。求最少花费。 | 将土地按w排序，维护一个高度递减的单调栈。不在栈中的土地存在与否不会影响答案。  设f[i]表示前i块土地的最少花费，f[i]=Min{f[j]+ h[j+1]\*w[i]}然后斜率优化来做即可。 | O(NLogN) |
| CF 241B | Friends | 给定n (<=50000)个数，选出m (<=n(n-1)/2)对数，使得每对数异或值的和最大。 | 按二进制建trie，枚举第m大的每一位是多少，通过判断每个数可以找到几个当前位和自己不同的数来确定。如果这位是1，继续做，否则把前面几位固定，这一位是1的所有异或值加起来。如果固定一个数，另一个数就都在一个子树里面。可以预处理每个子树第i位是1的有几个，这样即可在O(30)的时间内计算。 | O(n\*30\*30) |
| CF 238D | Tape Programming | 定义一个语言，其程序是一个由数字和”<”,”>”构成的非空串。从左往右扫，遇见数字便输出，非0就减1，否则删掉；否则就改变方向。如果连续两个都不是数字，则删掉先前的那个。如果跑出串外，则程序结束。给出m个询问[l,r]，问以这一段为程序各个数字被输出几次。 | 跑一遍主程序，如果跑到位置0，则再往右继续跑。记录每个位置第一次访问的状态，以及第一次从右边访问的状态。输出的时候比较一下是l-1先被从右边访问还是r+1先被从左边访问，输出那个状态减去l的状态即可。 | O(N\*10) |
| CF 317C | [Balance](http://codeforces.com/contest/317/problem/C) | 给定一张无向图，有n(<=300)个点每个点初始有a[i]升水。每个点上限都是v升。水可以通过边运输。给出b[]数组，求合法的运输方案使得点i(1<=i<=n)的水变成b[i]，并且步骤长度<=2\*n\*n。 | 仅需保留每个联通块的任一生成树。每次构造使得叶子节点满足条件，然后删掉。重复做n次。这样步数显然不会超过2\*n\*n。 | O(N\*N) |
| CF 305E | Playing with String | 给定长度为n(<=5000)的字符串，规定每次只能选定一个奇数长度且长于2的回文子串，将原串分为3串，分别是1到回文串中心-1，回文串中心和回文串中心+1到n。对分出来的每个串都可继续操作。现在两个人轮流操作，不能操作的输，问哪方必胜？若是先手，输出第一步，多解则输出最左边的。 | 先求出以每个位置为中心的最长回文长度。如果>2，则是染上白色，否则染上黑色。把连续的白点作为一段，则整个游戏的sg值就是每一段的sg值异或和。对于长度为n的一段，我们枚举一步操作之后左边的长度，记为i，然后这时的sg值就是sg[i]^sg[n-i-3]，然后就可以算出sg数组了。 | O(N\*N) |
| CF 319D | Have You Ever Heard About the Word? | 给定一个长度为n(n<=50000)的字符串，每次将长度最短的不重叠的重复子串删得只剩一个。求最后的字符串。 | 首先最多只有sqrt(N)中长度((1+x)\*x/2<=n)。所以直接模拟。枚举长度，每次用O(n/len\*logN)的时间判断是否可以删，然后用O(n)的时间删。可以用hash实现。判断是否可以删可以枚举n/len个点，求出相邻两点的lcs和lcp，它们之和如果大于len就可以删。 | O(sqrt(N)\*N+N\*log^2N) |
| CF 277D | Google Code Jam | Vasya在做google code jam。有n(<=1000)道题，每题有两部分分数easy和hard，Vasya读题不需要时间，做easy肯定能1A，耗费ts(整数)时间，得到ss分数，做hard有p的概率fst，耗费tl(整数)的时间，如果A了则获得sl的分数。现在他有t(<=1560)的时间。求最大的期望得分和对应的尽可能小的期望罚时。 | 首先对于一个做题序列，我们考虑怎样排序最优。通过比较交换相邻两项可以得出，应该按照tl \* p / (1 – p)排序最优。于是将所有题目排序，f[i][j]代表前i道题用j的时间的最大期望得分和最小期望罚时，然后类似01背包转移即可。 | 时间O(N\*T)  空间O(T) |
| CF 319E | Ping-Pong | 在这个问题中，每个时刻您都有一个区间的集合。您每次可以从集合中的区间(a, b)移动到另一个满足c < a < d或者c < b < d的区间(c, d)。您需要判断是否有一种从区间x到区间y的移动方案，或者添加一个区间(x,y)，并且保证区间长度严格递增。操作个数<=100000,坐标范围绝对值<=10^9，为整数。 | 对于两个区间，如果可以相互转换，则连双向边，否则连单向边。这步可以用线段树实现。对于双向边用并查集维护。判断时先判断能否只走双向边，不行的话如果可以走单向边到达，则目标区间所在并查集肯定有区间完全包含初始区间。 | 时间O(N\*31)  空间O(N\*31) |
| CF 314E | Sereja and Squares | X轴上给定n(n<=10000)个点，第i个点x坐标为i。现在要给这些点两两配对，并以它们的连线为对角线画正方形，并把坐标小的标为非x的小写字母，右边的标为对应的大写字母。要求正方形都不能相交。现在某些字母被擦掉了，只留下了部分小写字母，求有几种方案。 | 把小写当成-1大写当成+1，则任意时刻都不可以出现前缀和<0.令f[i][j]代表前i个点，前缀和为j，则f[i][j]=f[i-1][j-1]+f[i-1][j+1]。然后优化常数：  首先用i时刻的前缀和奇偶性和i是一样的。于是我们只存一半的数。然后注意到当i>n/2的时候某些状态最后是不可能转移到合法状态的。这样做就可以过了。 | 时间:O(N^2)  空间:O(N) |
| CF 235E | Number Challenge | 给定a,b,c，求：  其中d(n)代表n的约数个数。  1 ≤ *a*, *b*, *c* ≤ 2000 | 然后枚举k,  用莫比乌斯函数化掉gcd，然后枚举i,j。 | 时间  O(a^2loga) |
| CF 303D | Rotatable Number | 给n和x，要找1<b<x的最大b，使得存在b进制的类似142857的数。即所有从它通过旋转得到的数都是它自己乘以1,2,3...,6（从1到数的长度），且允许有前导零。1 <= n <= 5 \* 10^6，2 <= x <= 10^9 | 结论题:n+1要为质数,b是n+1的原根(n>1)。  参考：http://en.wikipedia.org/wiki/Cyclic\_number | O(N\*log^2N) |
| CF 241D | Numbers | 给定一个n(<=50000)的排列和质数p(<=50000)，要求删掉某些数，使得结果满足所有数的异或和等于0，并且按十进制从前往后依次无间隔地写在一行形成一个十进制数，这个数将会被p整除。输出任意解，或者No。 | 只要保留1到31的数，这样对于任意的p，有很大的概率有解。因为总共有2^31种情况，对于异或和为0的有2^31/31种，而这么多种情况对于mod p的值都是随机的，没有取到0的概率几乎为0((49999/50000)^(2^31/31)). | 时间  O(32\*32\*N)  空间  O(32\*32\*N) |
| CF 301C | Yaroslav and Algorithm | 有一种奇怪的语言，你可以用这个语言写一些奇怪的算法。这个算法接受一个字符串a作为输入。每条命令格式如下：s[i]>>w[i]或s[i]<>w[i]，含义为：如果s[i]在a中出现过，则将s[i]在a中第一次出现的位置替换为w[i]，区别是后一条命令执行完会终止程序。执行命令的规则如下：每次从上到下遍历每条命令，找到第一条可以产生效果的命令，执行之，然后再从头找新的命令。如果没有可执行的指令，算法结束。结束时的串就作为输出。  现在给出n个正整数，每个数不超过10^25，让你用这个语言写一个算法，要求这个算法对于每个给出的数作为输入，都能输出它+1的值。命令条数不超过50，对于每个输入算法的执行步数不超过200。n≤100。 | 对于所有数据，答案都是一样的。  ??0>>0??  ??1>>1??  ??2>>2??  ??3>>3??  ??4>>4??  ??5>>5??  ??6>>6??  ??7>>7??  ??8>>8??  ??9>>9??  ??>>?  0?<>1  1?<>2  2?<>3  3?<>4  4?<>5  5?<>6  6?<>7  7?<>8  8?<>9  9?>>?0  ?<>1  >>??  "?"会作为一个“这个地方需要+1”的标志。 | O(1) |
| CF 323C | Two permutations | 你有两个各包含n个元素的排列p和q，和m个由l1，r1，l2，r2组成的询问。每次询问在p中位置在[l1,r1],在q中位置在[l2,r2]中的数的数量。  1<=n<=1000000 1<=m<=200000 | 裸的主席树。 | 时间  O(NlogN)  空间  O(NlogN) |
| CF 269D | Maximum Waterfall | 给定n(<=10^5)条二维平面上的水平线段，保证两两不相交。最顶上和最下面有一条(-10^9,10^9)的线段。求一条从最上到最下的由线段组成的路径，要求高度递减，并且任意相邻两条线段没有高度在它们之间且投影到x轴上与它们都相交的线段（端点允许相交），在此基础上求所有相邻两条线段相交长度最小值的最大值。 | Dp.用一个set维护还没有被覆盖的线段。每加入一条线段先求出这一段里面有那几条线段，然后判断能否转移，然后就将这一段覆盖。 | 时间  O(N\*logN)  空间  O(N) |
| GCJ 2009 Final C | Doubly-sorted Grid | 一个长方形的网格每个格子里有一个小写英文字母。我们称这个网格是双排序(doubly sorted)的，当每一行从左到右都是不下降的，每一列从上到下也是不下降的。求有几种n行m列的双排列网格。  N,m<=10 | Dp.字母与字母之间肯定可以通过从矩阵下方到上方的一条轮廓线隔开。用f[i][j][k]代表i这题轮廓线左上方用不超过j的字母填充，且第k列之后这部分没有出现j字母的方案数。转移就是考虑k这列最下面要不要填j。要的话就把轮廓线往上推一格，从这转移。否则从k-1转移。 | 时空间  O(C(2N,N)\*26\*N) |
| CF 261E | Maxim and Calculator | 一开始有两个数a,b，初始是1,0，每一次你可以++b,或者a\*=b。给定l,r,p,求[l,r]之间有几个数x可以在p步之内使得a=x。  2<=l<=r<=10^9, 1<=p<=100 | Dp.首先10^9以内最大质因数<=100的只有3000000个。我们只考虑这些数。F[i][j]代b=i时，第j个数最少的步数-i。每次f[i][j]由f[i][Map[a[j]/a[i]]]和f[i-1][j]转移，其中Map[i]=j|a[j]=i。可以压掉第一维，加速转移。 | 时间  O(3000000\*100)  空间  O(3000000) |
| CF 338D | GCD Table | 有一个n行m列的表格G，G[i][j]=gcd(i,j).给你一个正整数数列a[1],a[2],...,a[k]。如果这个序列在G的某行的连续位置出现，则这个序列在表格中出现。询问序列a是否在G中出现。  n,m<=10^12,k<=10000,数列a中的每一个数都不超过10^12。 | i最小是lcm(a[1],a[2]…)  故令i=lcm。然后就得到一系列关于j的同余方程。  j = -k (mod a[k + 1])  解之得到任意一个j，判断是否合法。是则为yes，否则为no。因为所有合法的j都是j0+i\*k。注意乘法时可能会爆longlong。 | 时间  O(K \* log 10^12)  空间O(K) |
| CF 309D | Tennis Rackets | 给定一个正三角形，每条边有n个点(不包括三角形顶点)，将线段分为n+1段。  现在要在每条边上各选出一个点，组成一个钝角三角形，且每条边靠近顶点的m个点是不能用的。求有几种方案。  1≤n≤32000，0≤m≤n/2。 | 暴力+常数优化。。。 | 时间  O(N^2)  空间  O(1) |
| CF 266E | More Queries to Array... | 给n个数a[1],a[2]…，有两种操作：  给定l,r,x，将[l,r]内的数赋为x。  给定l,r,k，求[l,r]  1≤n≤100000 | 线段树。  将式子ai后面的那一项拆开(1-l为一个整体)，把i和ai当成常数，合并同类项，就可以发现只要知道ai\*i^0,ai\*i^1,ai\*i^2… ai\*i^k就可以算了。于是用线段树维护，合并的时候用类似方法即可。 | 时间  O(N\*logN\*25)  空间  O(N) |
| CF 273E | Dima and Game | Dima在纸上写下n对整数(l[i],r[i])(1<=l[i]<r[i]<=p)。然后玩家轮流进行操作。轮到自己时，可以进行下面的操作： 　　1.选择第i对数(1<=i<=n)，满足r[i]-l[i]>2; 　　2.将第i对数替换为( l[i]+floor((r[i]-l[i])/3) , l[i]+2\*floor((r[i]-l[i])/3) )或者( l[i] , r[i]-floor((r[i]-l[i])/3) )。floor(x)表示向下取整。不能进行操作的玩家则输。输出先手必胜方案数。  1<=n<=1000,1<=p<=10^9。 | 如果p<=1000000，那么就是sg的模板题了。但是p很大。打一个表可以发现，输出被分为了若干段，每一段都是同一个数。所以对于相同的x,y我们可以求出最大的满足i/3=x && i-i/3=y的i。这样一直跳就可以了。段数显然是logN的。得到i可以用数学方法计算，也可以用二分。这里我用了二分。 | 时间  O(log^2P+N)  空间  O(N) |
| USACO Open 07 | Connect | 给定一张2\*C的点阵图和一些边。有不超过N(N<=50000)次操作，每次可能会增加或删除一条边，或者询问 (r1,c1)和(r2,c2) 是否连通。注意，这里连通指的是所经过的点不能超出c1列和c2列之间。任何时候，边仅会出现在相邻两点之间。 | 线段树维护四个角的连通性。 | 时间  O(N\*logC)  空间  O(N) |
| USACO 2012 Dec gold 2 | First! | 给n个字符串，你可以任意规定字母之间的字典序大小关系。输出那些字典序可能最小的字符串。 | 建trie树，dfs，并维护一张有向图。当到达一个单词时，判断是否为拓扑图即可。dfs的时候对图的记录可以用位运算，即对于每个字母用一个int记录大于它的字母有哪些。 | 时间  O(字符串总长度+n\*26^2)  空间  O(字符串总长度+n\*26) |
| CF 293E | Close Vertices | 给定一棵n个点的树和W,L，每条边有权值，求有几对点，它们的距离<=L且权值之和<=W。  1<=n<=100000,1<=L<=n,1<=W<=10^9 | 裸的点分治。 | 时间  O(NlogN)  空间  O(N) |
| GCJ 2009 Final D | Wi-fi Towers | 平面上给定n个圆，每个点有一个点权（可能正可能负）。如果选了一个点，在它以内的所有点都要选。求所选的点权和最大值。  n≤500 | 裸的最大权闭合子图。  但是边数可能很多。  可以发现如果a->c,b->c，且角acb<=60°，那么长边不需要连。那么这样每个点入度<=6.边数少了很多。但实际上这样做常数很大，所以变成没有优化过的才能过了。 | 时间  O(N\*N\*N\*N)  空间  O(N\*N) |
| CF 261D | Maxim and Increasing Subsequence | 求数列a的最长上升子序列。  　　数列a给定如下：  　　1、数列的长度为n\*t；  2、ai=b((i-1) mod n)+1 (1≤i≤n\*t)，其中运算x mod y表示x除以y所得的余数。  1≤k≤10，1≤n,maxb≤105，1≤t≤109，n\*maxb≤2\*107 | t<=min(maxb,n)  直接暴力。  F[i]代表最后的数<=i时的最长长度。  直接暴力更新即可。  每一项只会被赋值maxb次。所以复杂度有保证。 | 时间  O(N\*MAXB)  空间  O(N) |
| CF 235D | Graph Game | 求环套树的随机化点分治期望复杂度。  3 <= n <= 3000 | 考虑(u,v)作为1的贡献，其中u作为选中的“重心”，v是这时候和u连通的一个点。  如果路径（u,v）没有经过环，那么贡献的概率是1/dis(u,v)，否则就是两条路径的贡献和再减掉并的贡献。 | 时间  O(N^2)  空间  O(N^2) |
| GCJ 2009 Final B | Min Perimeter | 求最小周长三角形  N<=100000 | 类似平面最近点对。分治，合并的时候，只扫p\*p/2大小的矩形内的点。 | 时间  O(NlogN)  空间  O(N) |
| CF 267C | Berland Traffic | 给定一张无向图，每条边有容量，要求  构造一个网络，使得对于任意x,y,任选一条从x到y的路径上边的流量和（注意有正负之分）相等，并且边上流量<=容量。  N<=100,m<=5000 | 高斯消元。  不妨设1到i的路径流量和为x[i]，则一条边<u,v>的流量就是x[v]-x[u]。以此列方程即可。 | 时间  O(N^3)  空间  O(N) |
| CF 306D | Polygon | Polycarpus喜欢凸多边形，尤其喜欢每个角角度都相同，每条边长度都不相同的凸多边形。你需要在知晓给定的顶点数的情况下，为他描绘出这样的任意一个凸多边形。  3<=n<=100 | 以(0,0)为第一个点，维护一个向量。向量每次旋转一个外角，加长一点点长度，然后把向量作用到上一个点，得到下一个点。最后一个点直接求和x轴的交点即可。 | 时间  O(N) 空间  O(1) |
| CF 253E | Printer | 有n个任务，任务i在t[i]时刻出现，优先级为p[i]，要用s[i]的时间。任务可以拆开来做（不可拆分为小数时间）。现在有一个机器，从时刻0开始工作，每次会选择队列里面优先级最高的来做。任务中有一个的优先级未知（p[i]=-1），但是完成时刻是知道的。其他任务优先值已知，完成时刻未知。求一组合法解。保证有解。  n≤50000 | 二分优先级。模拟。 | 时间  O(N\*logN)  空间  O(N) |
| USACO March Contest 2009, Gold Division, Problem 2 | Cleaning Up | 有n个数，现在要将它们分成几个子串，费用为每个子串内不同数字的平方和。求最小费用。  1≤m≤n≤40000  M是数字种类。 | Dp。观察到答案<=sqrt(n)，所以我们维护sqrt(n)个数，第j个数代表从i到哪里分为一组，花费j^2的费用。直接转移即可。 | 时间  O(N\*sqrt(N))  空间  O(N) |
| CF 316D | PE lesson | 对于一个排列，每次可以任意交换两个数，但第i个位置不能交换超过a[i]次（a给定，1<=a[i]<=2）。求有几种的n的排列，通过上述操作后可以变成一个有序排列。  N<=10^6. | 令1的个数=a；2的个数为b观察什么样的排列是合法的。发现把排列拆穿若干个置换，如果每个置换中1的个数<=2，那么它就是合法的。如果b=0，则可以用dp来计算。F[i]=f[i-1]+(i-1)\*f[i-2]。令ans=f[a]  多一个2，放在哪个置换，置换里的哪个位置都是可以的。于是答案就是ans\*(a+1)\*(a+2)\*…\*(a+b) | 时间  O(N) 空间  O(N) |
| USACO Jan 2011 | Bottleneck | 给定一棵n个点的树，1为树根，点i有c[i]头奶牛，点i和i的父亲间的边权为m[i]，代表每个单位时间这条边只能通行m[i]头奶牛。只要符合边权的限制，每头牛在每单位时间可以无限往上走直到根。现在给定K个询问，每个询问给定T[i]，询问在T[i]的时间内能有几头奶牛走到根。  1≤n≤100000，1≤k≤10000，1≤Ti≤109，1≤Ci≤109，0≤Mi≤109， | 可以发现，每条边在都会在某个时间后不会再满流。在那之后我们便把这条边连接的两个点合并。变成另一棵等价的树。合并的话就是c[i]+=c[j],inflow[i]+=inflow[j]-m[j]。  于是我们求出每条边的时间，用一个堆来实现，每次取出最小边来合并。  对于每条边都满流的树，答案就是c[1]+inflow[1]\*T。 | 时间  O(NlogN+KlogK)  空间  O(N) |
| CF 321D | Ciel and Flipboard | 给定n\*n(n为奇数)的矩阵，每个位置上有一个数。每次可以选择一个边长为x=(n/2+1)的正方形，将里面的数\*=-1。要求最大化各数和。  N<=33 | 每一行相差x的两个数和那一行中间那个数三个直接一定会有偶数个数被翻转。  于是枚举第x列的前x行个数。对于x行的前x-1列，它们互不相关，所以贪心的选取最优值即可。 | 时间  O(2^x\*x\*x)  空间  O(x) |
| CF 325d | Reclamation | 有一个r\*c的地图，把左边界和右边界粘起来使得形成一个圆柱，现在要不断地挖去其中的格子，要求任何时候都存在一条从最上方到最下方的路径(四联通)，如果某次操作不满足要求则不做，问最后有多少次操作是成功的。  r,c<=3000，n<=300000。 | 把环拉成链，对于一个询问(i,j)如果(i,j)能够通过放置的格子走到(i,j+c)（八连通）那么就是非法的。然后用并查集实现即可。  注意，即使拉成链也应考虑环。 | 时间  O(r\*c+n)  空间  O(r\*c) |
| CF 266D | BerDonalds | 给定一个无向带权联通图，求一个点（可以在边上），使得它到所有点的最大距离最小。  距离定义为两点间的最短路。  n<=200，c<=100000，没有重边或自环。 | 枚举边，可以发现每个点到当前边上某点的距离是个斜率为1和-1的分段函数。然后求出它们的靠上的交点，更新答案即可。 | 时间  O(N^3)  空间  O(N^2) |
| CF 263E | Rhombus | 给定n\*m的非负整数表和k。定义f(x,y)=(k<=x<=n-k+1且 k<=y<=m-k+1)  求f(x,y)的最大值。并输出x,y  1<=n,m<=1000 | 维护a[i][j]代表以i,j为中心的正方形的左上角三角形数值和。然后知道f(x,y)后就可以O(1)求得f(x,y+1)。  依此类推即可。 | 时间  O(N\*N)  空间  O(N\*N) |
| GCJ 2014 Final D | Paradox Sort | 给定T组数据，每组数据给定A和一张n个点的竞赛图。如果i连向j，则代表i相对j优。求一个字典序最小的排列，使得按照这个顺序比过去，每次保留最优的，最后能够得到A。  无解输出IMPOSSIBLE  1<=N,T<=100，0<=A<100 | 判断是否有解可以从A开始dfs，如果能够遍历所有点，则有解。然后我们一位一位确定。枚举第一位，判断是否可行，依此类推。  判断的方法如下：  如果这一位是A，则通过模拟判断。  否则先将已确定的部分模拟一遍，得到结果记为B,然后将这些点删除。然后从A开始dfs。对于没有遍历到的点必须都比B来的劣。 | 时间  O(?)  空间  O(N\*N) |
| CF 346E | Doodle Jump | 有n个平台。第x(1≤x≤n)个平台的高度是a·x mod p，其中a和p是互质的正整数。问给n个数排序后相邻两项差的最大值是否<=h。t组数据。  1≤t≤104，1≤a≤109, 1≤n<p≤109, 0≤h≤109 | 定义f(a,n,p)为差的最大值。则f(a,n,p)=  long long t = a \* n / p;  if (a \* n < p) return std::max(a,p - a \* n);  if (a \* n % p < a \* (p / a - 1)) --t;  return f(std::min(p - p % a,p % a),t,a); | 时间  O(logn)  空间  O(logn) |
| CF 342D | Xenia and Dominoes | 给定一张3\*n的网格。你要用多米诺骨牌铺满它。网格中有某些格子是禁止的。还有一个特殊的格子，上面也不能放骨牌。要求最后的方案至少有一个骨牌能够通过特殊方格移动。  求方案数。  3<=n<=10^4 | 枚举可以移动的骨牌的放置情况，然后状压dp。最后用容斥原理合并。 | 时间  O(2^6\*n)  空间  O(2^3\*n) |
| CF 341E | Candies Game | 给定n个数，每次可以任选两个数x,y(x<=y)，得到新的(x’,y’): x’=2x,y’=y-x。求一种方案使得最后只有两个非0的数。  N<=1000 | 对于三个非0的数a,b,c(a<=b<=c)，我们可以通过特定操作使得a减小：  令q=b/a。类似快速幂，如果q的这位是0，则c-=a,否则b-=a。  这样b最后就是b%a<a。 | 时间  O(n\*logn)  空间  O(n) |
| CF 257E | Greedy Elevator | 要求你模拟电梯运行。  规则如下，共有m层和n个人，每个人在ti时刻在si层等待，目的地是fi层。电梯0时刻在1层。如果当前电梯内没人，则电梯不动，否则如果想要电梯上去的人数>=下去的，则下一时刻电梯往下走一层。假设人进出电梯是瞬间完成的。  1<=n<=100000，2<=m<=100000，1<=ti<=1000000000，1<=si,fi<=m，且si≠fi | 用两个set维护电梯里外的人，关键字是楼层。然后模拟即可。 | 时间  O(N\*logN)  空间  O(N) |
| CF 305D | Olya and Graph | N个点，m条边的有向图。要求添加若干条边，使得图满足如下性质：  1、 从点i出发，可以到达点i+1，i+2,…..,n。 2、 任意从u到v的有向边满足不等式：u<v。 3、 两点之间最多有一条边。 4、 对于一对点i、j(i<j)，若j-i<=k，那么从i到j的最短距离等于j-i条边。 5、 对于一对点i、j(i<j)，若j-i>k，那么从i到j的最短距离等于j-i或j-i-k条边。  求方案数。  2<=n<=10^6,0<=m<=10^5,1<=k<=10^6 | 可以发现总共就只有两种边。。。然后第一种边是(I,i+1)第二种是(I,i+k+1)。第一种边必须填满，第二种边两起点最大跨度不能超过K+1.然后直接统计就行了。 | 时间  O(N)  空间  O(N) |
| CF 273D | Dima and Figure | 在n\*m的长方形方格里面选取一个联通块（四联通），要求任两点的最短距离等于它们的曼哈顿距离。  求方案数。  1<=N,M<=150。 | 可以发现每一行每一列都最多只有一个联通块。  然后可以dp。  F[i][j][k][l]记录前i列中，第i列的放置(j,k)的方块，l代表之前是否有方块高于j，是否有方块低于k。  然后维护二维前缀和即可转移。 | 时间  O(N\*M\*N)  空间  O(N\*M\*N) |
| CF 264E | Roadside Trees | 有n个位置可以种树。共m个操作，每次可能是种一棵树，或者砍掉一棵树。每次操作后树都会长高一个单位长度。要求每次操作后都要输出当前高度的最长上升子序列。保证树都在不同位置，并且高度都不同，而且砍掉的位置不会再种树。  1≤n≤10^5；1≤m≤2·10^5，种的树初始高度<=10，砍的树都是前10棵之一。 | 维护x轴和y轴的线段树，值就是以这棵树为开头的最长长度。每次操作都将x轴或y轴的前10棵树取出，再按顺序插入即可。 | 时间  O(N\*10\*log^N)  空间  O(N) |
| CF 325E | The Red Button | 有0到n-1共n个点，点i可以连向i\*2%n和,i\*2+1%n。求一条从0开始的哈密顿回路。  N<=10^5 | 首先n如果是奇数则无解。证明可以考虑0和n-1的前一个数。  然后有个结论：要么是x->2x,x+n/2->2x+1，要么是x->2x+1,x+n/2->2x。  那么我们先按照上面的规则随意连。这时候会出现若干个圈。如果>=2，可以证明肯定存在x满足x和x+n/2不在同一个圈（证明可以考虑反证，如果不满足则说明只有一个圈）。然后我们就可以合并了。  用一个dfs来解决。 | 时间  O(N)  空间  O(N) |
| CF 294D | Shaass and Painter Robot | 给定n\*m的网格，一开始都是白色的。一个机器人一开始从x,y往左上左下右上右下移动。碰到边界就按照光的反射定律改变方向。被机器人走过的格子被染成黑色。如果某一时刻网格呈黑白相间，机器人停止移动。求机器人走了多长。如果无法停止输出-1.  N、M<=100000 | 把整个网格看成好几条斜线交错在一起。机器人必须走完所有斜线。于是我们只记录起点和终点，模拟机器人移动就可以了。如果移动次数超过2\*点数，则无解。 | 时间  O(N) 空间  O(N) |
| CF 258D | Little Elephant and Broken Sorting | 给定一个初始排列。长度为n。给定m对数(ai,bi)，代表有1/2的几率交换ai,bi。问按照读入的顺序交换，最后期望有几个逆序对。  1 ≤ n, m ≤ 1000 | f[i][j]代表p[i]>p[j]的概率。  交换a,b：  F[a][b]=0.5  F[i][a]=f[i][b]=0.5\*f[i][a]+0.5\*f[i][b]  F[a][i]=f[b][i]=1-f[i][a] | 时间  O(N\*N) 空间  O(N\*N) |
| US open 10 | Triangle Counting | 给出n个笛卡尔坐标系上的整点，统计有多少三角形包含原点(0,0)。  1<=n<=10^5  -10^5<=X\_i,Y\_i<=10^5  不会有两点的直线经过原点，特别地，不会有点恰好是原点。 | 正难则反，统计非法三角形，再用总数减掉。将所有点极角排序，按顺序扫描，同时维护以当前点和原点组成的平面上的点的个数x。以当前点为最左边的非法三角形个数为C(x – 1,2)。 | 时间  O(nlogn)  空间  O(n) |
| CF 332E | Binary Key | 假设有p和q两个长度为正整数的字符串。其中，串q只包含字符0和1。q串是循环的无穷的。q串上为1的所对应的p串上同一位置的字符，如果不是空，我们将其取出。我们按顺序一个个取出，最后连在一起，就是q在p上的作用结果。  现在给定结果s和p以及q的长度k，求任意一个合法的q。  k <= 2000, |p| <= 10^6, |s| <= 200 | 枚举q串1的个数。这样我们就能知道每一个1对应的s上的字符串。  而p上q中每个1所对应的字符串也是可以算的。  于是从后往前确定是否可以为1即可。 | 时间  O(k^2)  空间  O(|p|) |
| CF 332D | Theft of Blueprints | 给出一个n个点的带权无向图，满足对于任意一个大小为k的顶点集合S，恰好有一个点与S每一个点都有边。令这个点为v(S)，并且对S进行操作的代价是S中每个点与v(S)的边权之和。现在求对于一个大小为k的子集操作代价的期望。  1<=k<n<=2000 -1<=c\_i,j<=10^9 | 分子分母分开计算。对于确定了一个点和一条边，这条边此时贡献是可以计算的。为边权\*C(这个点的度数,k-1)  预处理组合数即可。 | O(nk) |
| CF 251D | Two Sets | 给定n个数，要将其划分成两个集合（可以为空集），记两个集合的异或和为x1,x2，求x1+x2的最大值，并且要求最小化x1 | 贪心，从高到低确定x1,x2的状态，用高斯消元来判断是否可行。 | O(63^2\*N) |
| GCJ 2014 Final F | ARAM | 有n只英雄，每一只的胜率已知。一开始会随机分配到一只英雄，你可以选择重新随机（Rerolling），但是需要消耗1元，一场比赛中可以Rerolling多次。一开始有R元，一场比赛结束后会得到1/G元。但任何时候总数不能超过R元。你现在要玩10^100局，求最优策略下，期望的取胜次数的比例。  n≤1000,R,G≤20 | 二分答案，A[M]代表以M元开始玩，直到钱攒到M+1元期望赢的场数。  M<1:  1<=M<R:  M=R:  如果存在i使得A[i]>=0，则这个答案是可行的。 | 时间  O(R\*G\*n\*logn)  空间  O(n+R\*G) |
| USACO Dec10 Gold | Threatening Letter | 给定a串和b串，保证只含大写字母。现在可以从a上面复制任意多个子串。问最少复制几个才能将它们首尾相接得到b串。  |a|,|b|<=50000 | A串建后缀自动机。然后贪心往下匹配即可。 | 时间  O(N\*26)  空间  O(N\*26) |
| GCJ 2011 Final A | Runs | 给定长度<=450000的字符串。我们定义它的|run|=极大的且由相同字符组成的子串个数。  现在将这个字符串重排列，问有几种方案使得|run|和原来相等？  保证|run|<=100 | Dp:f[i][j]代表前i个字母，组成的run=j。决策是枚举i这个字母分成几段插入，并且有几个插在不同字母之间，有几个插在相同字母之间。  需要预处理组合数。 | 时间  O(100^3\*26)  空间  O(450000\*100) |
| CF 241E | Flights | 给定一张n个点，m条边的有向图，边权都为1，对于边u->v，保证u<v。现在你可以选择某些边将边权改为2，要求任一从1到n的路径长度都相等。  2 ≤ n ≤ 1000，1 ≤ m ≤ 5000 | 差分约束系统。  注意要去掉1无法到达以及无法到达n的点。 | 时间  O(K\*M)  空间  O(M) |
| USACO Open 13 | Figure Eight | 给定一个n\*n的网格，某些位置是缺损的。现在要在上面刻出一个8，8由两个矩形组成，要求上面的矩形底边被下面矩形的顶边包含。并且缺损的位置不能用。求上下两个矩形面积乘积的最大值。  N<=300 | Dp.  先预处理出up[k][i][j]代表以第k行为底边，并在第i到j列之间的矩形的面积最大值。  然后就可以直接算了。 | 时间  O(N^3)  空间  O(N^3) |
| CF 241F | Race | 给定n\*m的网格，里面有若干条横向或纵向的道路，还有两条道路相交形成的交叉路口。道路占据连续的横向或纵向的若干格子。剩下的格子都是障碍，用’#’表示，交叉路口用’a’到’z’表示，并且唯一。其他的都是一些数字，代表进入这个格子需要多久才能走出去，而交叉路口需要1的时间，障碍无法进入。  现在给定起点和终点，以及之间经过的交叉路口，求第K分钟的位置。  N,m<=100,K<=100000,经过的交叉路口数<=1000（即可能重复通过某个位置） | 模拟。 | O(N\*M\*1000) |
| CF 286E | Ladies' Shop | 给定n个正整数ai和m,要求给出k个正整数pi,满足从这些数中任意选取若干个数（同一个数可以选取多次）相加和组成的集合除去>m的数后恰好就是集合a，并要求K最小。如果无解输出NO。  1 ≤ n, m ≤ 10^6，1 ≤ a1 < a2 < ... < an ≤ m | FFT.  首先p肯定是a的子集。  我们选取a中不能被表示成ai+aj的数即可。如果有某个数可以被表示成ai+aj但却不属于a，则无解。 | O(M\*logM) |
| CF 338E | Optimize! | 给定n,len,h,a[],b[]优化下面代码  1 ≤ len ≤ n ≤ 150000  1 ≤ a\_i,b\_i,h≤ 10^9 | 线段树。  Ai=h-ai,然后问题就变成是否能够找到一个匹配，使得ai<=bi。将bi的位置打上+1,ai的位置打上-1，如果有哪个后缀和<0则不能。这个可以用线段树来做。 | 时间  O(N\*logN)  空间  O(N) |
| CF 293B | Distinct Paths | 小美有一个n\*m的木板，一些块已经被涂上给出的k种颜色中的一种。你需要把每个没涂色的块涂色使得从左上角到右下角的每条路径都不会经过两个颜色一样的块。路径只能向右或向下走。  输出答案%1000000007。  1<=n,m<=1000,1<=k<=10 | 爆搜，某些状态是同构的，通过这个来剪枝。 | 时间  O(?) |
| CF 348E | Pilgrims | N个点，有M(<N)个黑点，其余是白点。现在要选择删掉一个白点，让最多的黑点到所有距离它最远的黑点都不连通。并输出方案数。  n<=100000 | 每个黑点到距它最远的黑点都要经过最远黑点对的中心。并且要满足一个黑点，可以删掉的白点刚好是一条链。于是我们把所有链标记上即可。 | 时空间O(N) |
| CF 235C | Cyclical Quest | 给一个字符串S，再给一个字符串T,设T的长度为len，问T的循环串在S中出现的次数，这里循环串的定义是：对于一个长度为len的字符串，我们把它首尾相接，然后从任意位置开始走len步所得到的串我们叫做T的循环串。如abaa的循环串有 abaa，baaa，aaab，aaba。（注意如果重复只算一次。比如aaa的循环串只有一个aaa） | 后缀自动机。。。  这题就是求几个串的出现次数罢了。。。  所以把S建成sam，然后把T \* 2变成T',拿T'在sam上面跑。  维护一个匹配长度。如果超过T的长度就要减成T的长度，然后记得看一下当前这个串是否属于点p，也就是if l[fa[p]]>=len(T) then p = fa[p]; | 时空间  O(N\*26) |
| CF 331C | The Great Julya Calendar | 给定n，每次可以将n扣掉n一个数。这个数必须是n的某一位。重复执行这个步骤，问最少操作几次。  N<=10^18 | 每次肯定是扣掉最大的一位（因为答案肯定是递增的。这点可用归纳法证明）。F[i][j][k]代表i位数，除了最后一个是j，其他都是9，并且之前出现过的最大位是k，此时要将这个数扣到<=0的最少操作次数。然后直接转移就好了。 | 时间  O(18\*10\*10\*10)  空间  O(18\*10\*10) |
| usaco Dec 2007 gold | Best Cow Line | 给一个字符串s，每次可以取出s的头或尾，放到t串的后面。要求t的字典序最小。输出t。  S长度<=30000 | 后缀数组。贪心。 | 时间  O(N\*logN)  空间  O(N) |
| CF 249E | Endless Matrix | 有一个矩阵，里面有连续的数字，从1开始。 若a\_{i, j} < a\_{t, k} (i, j, t, k ≥ 1) 当且仅当 1.max(i, j) < max(t, k)  2.max(i, j) = max(t, k) 且 j < k  3.max(i, j) = max(t, k), j = k 且 i > t  给定 x\_1, y\_1, x\_2 和 y\_2 (x\_1 ≤ x\_2, y\_1 ≤ y\_2) 求以(x\_1,y\_1)为左上角，(x\_2,y\_2)为右下角。  t ≤10^4,  x\_1,y\_1,x\_2,y\_2≤10^9 | 转化成计算(1,1)到(x,y)的和。  会爆longlong，用二分乘法。 | 时间  O(T\*log10^9)  空间  O(1) |
| CF 339E | Three Swaps | 给定n的一个排列，已知它是由升序的排列通过不超过三次区间翻转得到的。  要求三次的l,r。  N<=1000 | 首先，一定可以找到等价操作使得每次操作l,r都满足abs(a[l-1]-a[r])==1 or abs(a[l]-a[r+1])==1.（因为如果答案<3，一定满足，否则如果相邻两次操作有包含关系，可以调换位置，然后可以对前两个操作分类讨论即可得证）  然后，用一个dfs就可以解决了。 | 时间  O(?) |
| CF 283E | Cow Tennis Tournament | 有n头奶牛，每头奶牛有一个能力值si，能力值互不相同。它们两两进行比赛，能力值高的胜。FJ会执行k次操作，每次给定a,b,对于任意两头奶牛x,y（满足sx,sy属于[a,b]），反转这场比赛的胜负。  求有几对(p,q,r)满足p赢q，q赢r，r赢p。  3≤n≤10^5;0≤k≤10^5,1≤si≤10^9,1≤ai<bi≤10^9. | 正难则反。我们计算非法的方案数。即有几对(p,q,r)满足p赢q,r。然后就转化成求每头牛的胜利场数。然后我们枚举奶牛i,维护一个bool数组，代表是否败给i。对于一次操作a,b，如果影响了i，我们将影响的区间异或1。这些可以用排序加线段树解决。 | 时间  O(NlogN)  空间  O(N) |
| Codeforces #152 (Div. 2) E | Piglet's Birthday | N个数ai(<=100)，代表第i个架子上蜜罐的个数，一开始里面都是满的。q次操作(ui,vi,ki)，每次在第ui个架子上随机选出ki个蜜罐，然后吃掉里面的密（如果有的话），并把空罐子放到第vi个架子。求前i次操作完了之后期望有几个架子上面的罐子都是空的。  1 <= ui, vi <= n, 1 <= ki <= 5,1 <= n,q <= 10^5 | Dp。  F[i][j]代表第i个架子上只剩j个满的罐子的概率。答案就是sigma(f[i][0])。然后转移就枚举剩余满罐子数，以及选到了几个满的。然后用组合数算下概率就可以了。 | 时间  O(N\*500)  空间  O(N\*100) |
| CF 238E | Meeting Her | N个点，m条有向边。有k条公交线路sk到tk，公交车会随机选择一条最短路。（如果找不到就不走）现在要从a到b，求最坏情况下最少乘车次数。无法到达则输出-1.  1 <= n <= 100,0≤k≤100  M<=n\*(n-1) | Dp  首先求出在公交路线上的点和一定在公交路线上的点。  Dp[i][j]代表在点i乘坐j（j==0则代表下车了）路公交车的答案。然后用类似拓扑排序（即度数为0才更新）和spfa的方法更新。 | 时间  O(N^3)  空间  O(N^2) |
| CF 249D | Donkey and Stars | 给定n个整点，给定两条射线(斜率为a/b c/d，角度为α1,α2)，每个点可以走到以当前点为起点的两条射线之内的点。  求从原点出发的最长路径。  1≤n≤10^5，0≤a,b,c,d≤10^5，0°≤α1<α2≤90°a+b>0且c+d>0，0≤x,y≤10^5 | 将坐标变换之后就是最长上升序列了。  变换的规则可以用和两条射线的叉积为坐标。 | 时间  O(N\*logN)  空间  O(N) |
| CF 274C | The Last Hole! | 给定n个圆，一开始半径都是0，t时刻半径为t。多个圆在某些时刻可能会围出一个未被圆覆盖的封闭区间。求出最早的时刻使得在此之后没有封闭的区间出现。如果至始至终都没有出现则输出-1.  0<n<=100, -10^4≤xi,yi≤10^4 | 最晚被覆盖的点一定是锐角三角形的外心或矩形的中心。然后暴力即可。 | 时间  O(N^4)  空间  O(N) |
| CF 301E | Yaroslav and Arrangements | Yaroslav称数列a1，a2，… ， ar为良好的，当它满足：  1.| a1 – a2 | = 1，| a2 – a3 | = 1，…，| ar-1 – ar | = 1，| ar – a1 |= 1  2.a1=min(a1,a2,…,ar)  Yaroslav称数列b1，b2，… ， br为优秀的，当它满足：  1. 数列中的元素不下降  2. 满足 1 ≤ r ≤ n 、 1 ≤ bi ≤ m  3. 通过重排数列中的元素可以得到至少一个至多k个不同的良好数列。  给定n,m,k,求有多少个满足条件的b数列。答案模掉10^9+7.  n，m，k≤100 | Dp  F[i][j][k][l]代表用了前i个数，b数列已经有了j个数，有k对相邻的数i，a数列的方案数是l的答案。  转移用递推的方式转移，每次枚举加几个i+1，假设加x个，然后这些数显然要填满k个空隙。于是a的方案就乘上C(x-1,k-1)。  答案=每个i<m且k!=0的状态方案之和。  时间应该是n^5，但是由于常数小，还是可以很轻松过掉，空间可以用滚动数组压掉一维。  至此问题得解。 | 时间  O(N^5)  空间  O(N^3) |
| Codeforces #158 (Div. 2) E | Dividing Kingdom | 给定平面上n个整点，现在要画两条和y轴平行和两条和x轴平行的直线将点分成9个集合。要求现在给定这九个集合每个集合的个数，求方案。直线不能过点。  9 <= n <= 10^5  1 <= ai <= 10^5  -10^9 <= x, y <= 10^9 | 爆搜出九个集合的个数，然后边搜边判断合法性。判断合法性可以用主席树来做。 | 时间  O(9!\*logN)  空间  O(NlogN) |
| CF 254D | Rats | 给定n\*m的网格，有些格子上有墙壁，有些格子上有老鼠，其余格子为空。现在要放2个炸弹，炸弹只能放在空的位置上。炸弹在0时刻能够炸掉所在位置的老鼠，下一个时刻可以波及到上下左右四个方向相邻格子，但是不能穿过墙壁，最多往外波及d次。求方案，无解输出-1。  4<=n,m<=1000，1<=d<=8 | 每个格子最多炸到2\*d^2+2\*d+1个格子。所以老鼠不会超过362只。我们预处理出每个格子引爆炸弹可以炸到的老鼠集合。然后枚举炸到第一只的炸弹的位置，再找到没有被炸到的老鼠，再找炸到这只老鼠的炸弹的位置，两者并起来看能否覆盖所有老鼠。 | 时间  O(K\*(362\*181+181\*362))  K是bitset的复杂度  空间  O(N\*N\*K) |
| CF 280D | k-Maximum Subsequence Sum | 给定n个数和m个操作，一种操作是将一个数重新赋值，另一种给定L,R,K，求在[L,R]之间最多选出K段不想交区间使得它们和最大。  1<=n,m<=10^5，序列元素绝对值不超过500，修改操作中的k不超过20。 | 通过费用流的思想，用线段树模拟增广，即找到最大子段和，取反，重复K次。 | 时间  O(N\*logN\*20)  空间  O(N) |
| CF 286D | Tourists | N对人,第i对人在qi时刻从(-1,0)和(1,0)出发，以y轴正方向行走，每秒一个单位长度。有m堵墙，第i堵墙在ti时刻出现在y轴的[li,ri]区间，并不会消失。求每对不能看到对方的时间。  1<=n,m<=10^5,0<=li<ri<=10^9,0<=ti<=10^9,0<=qi<=10^9 | 线段树。把墙拆成不相交的若干段（可以用线段树），然后一堵墙只会对qi在一段区间的游客有贡献，而且是呈等差数列。于是直接离线处理，最后排序扫描即可。 | 时间  O(NlogN)  空间  O(NlogN) |
| CF 243C | Colorado Potato Beetle | 有一个(1010 + 1) × (1010 + 1)平方米的土地，一开始人在地图中央喷洒杀虫剂，然后他会移动n次，每次会往上下左右四个方向移动x米，所经过的每个格子都会喷洒到杀虫剂。虫子会从地图的边缘侵入，以四联通方式拓展，但是无法到达喷过药的格子。求有几个格子没有被虫子入侵。  1≤n≤1000, 1≤x≤1000000 | 离散化+BFS。 | 时间  O(N\*N)  空间  O(N\*N) |
| CF 243D | Cubes | 3维空间中，在以(i,j,0),(i-1,j-1,0)(1<=i,j<=n)为对角的正方形上有aij块单位立方体。现在从无穷远的地方以v向量方向看去，问几个立方体没有被完全挡住。  1≤n≤10^3,  |vx|,|vy|<=10^4,|vx|+|vy|>0。0≤aij≤10^9,1≤i,j≤n | 将点投影到直线上面，然后转换成线段覆盖，然后就是经典的线段树问题了，只要维护区间最小值，标记下传即可。投影可以用叉积。 | 时间  O(NlogN)  空间  O(N) |
| CF 268D | Wall Bars | 要建造一个高度为n的wall bar，共有五根直立的杆子，其中四根组成正方形，另一根位于中央。高度为i时，从中央的杆子要恰往一个方向安装一根横向的杆子连向四根中的某一根。小孩从地板上可以爬到高度<=h的任意一根横杆，在横杆上的小孩可以爬到和它方向相同且高度差不超过h的横杆。现在要求小孩至少可以爬到>=n-h+1的高度，求方案数(mod10^9+9)  1≤n≤1000, 1≤h≤min(n,30) | Dp  F[i][j][k][l][x]代表处理到高度为i,其余三根杆离i的高度差为j,k,l,且i是否可达(x==1可达，否则不可达)时的方案数。  然后直接转移就可以了。 | 时间  O(N\*H\*H\*H\*8)  空间  O(N\*H\*H\*H\*2) |
| CF 264D | Colorful Stones | 给定两个场字符串s1,s2,保证只含R,G,B三种字母，一开始一个人位于s1的首字母，另一个人位于s2的首字母，现在可以发布若干个指令，每个指令为R,G,B之一，代表位于这个字母的人往前走一步。不能越界。求有几个状态可以到达，两个状态不同，当且仅当有某个人位置不同。  |S|<=10^6 | 假设指令序列为I，则A,C是I的子序列，B,D不是。  固定第一个人的位置，求出另一个人可以到达的位置的最左和最右（条件就是D不能是A的子序列，B不能是C的子序列）。但是中间会有一些位置无法到达，设第一个人的位置后两个字母是xy，则无法到达的位置后两个字母必定是yx(x!=y)。 | 时间  O(N\*9)  空间  O(N\*9) |
| CF 329D | The Evil Temple and the Moving Rocks | 要在n\*n的网格上放置石头，石头共有4种，有无限多块，每块石头只要被激活就会往一个特定的方向走（上下左右之一，取决于种类），直到边缘或碰到另外一块石头，并且此时会激活所碰到的石头。一块石头只要有移动就会发出声响，现在要求构造出一个方案，并激活一块石头使得发出声音次数>=x。输出方案和所激活的石头。  N<=300,k=n/2,x<=k^3-k^2 | 构造题。  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.>.v  ^.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<.<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<  这样重复k次输出即可。  发声次数大约在k^3。  上例是k=50的情况。其余类似。 | 时间  O(N\*N)  空间  O(N\*N) |
| CF 285E | Positions in Permutations | 我们称排列中的第i个位置是完美的，当且仅当|P[i]-i|=1。  请求出长度为n的而且完美的位置数刚好为k的排列数是多少。答案要求取模10^9 + 7。  1 ≤ n ≤ 1000,  0 ≤ k ≤n | 状压Dp+容斥  我们先放完美的位置，其余位置先不考虑。  f[i][j][k]代表前i个位置放了j个完美的，k属于[0,4)，代表i和i+1是否被用过。那么至少有j个完美位置的方案数就是f[n][j][0..4]\*(n-j)!。  再令F[i]为i个完美位置时的答案。  F[i]=f[n][i][0..4]\*(n-i)!-F[j]\*C(j,i)(j>i)其中C是组合数。 | 时间  O(N\*N\*2^3)  空间  O(N\*N\*2^2) |
| CF 333C | Lucky Tickets | 一个k-lucky ticket是一个8位整数（可以含有前导零），并且能够通过在数字之间穿插(,),+,-,\*使得结果为k。  给定k,输出m个k-lucky ticket。  0 ≤ k ≤ 10^4, 1 ≤ m ≤ 3·10^5 | 爆搜。  我们先爆搜4位，然后看剩下的能不能通过加减一个4位的数得到k，可以就加进set。 | 时间  O(爆搜)  空间  O(m) |
| USACO 2008 Open Gold | Cow Neighborhoods | 有n头奶牛，奶牛i位于(xi,yi)，两头奶牛有边当且仅当它们的曼哈顿距离不超过C。求有几个联通块，以及最大联通块的点数。  1 <= N <= 10^5, 1 <= C <= 10^9,1 <= x, y <= 10^9 | 先旋转坐标轴，旋转45度，然后扩大sqrt(2)倍。然后从左到右逐个添加点，并维护一个队列，保证里面的点的横坐标差<=C。并且队列的关键字是y坐标，这步可以用平衡树维护。将i和i的前驱后继连边。可以证明任意两个在同一块里的点都可以通过只走这两种边到达（可以用反证法证明，即假设第一次无法到达时i->k，则中间必定有j且j和k不同集合，但是j和k明显可以互相到达，矛盾）。 | 时间  O(NlogN)  空间  O(N) |
| CF 360D | Levko and Sets | 给定A,B数组和质数p，其中A有n个数，B有m个数。现在要生成n个集合，第i个集合一开始只有一个元素1，然后可以从集合里任意选出一个数c，对于(1<=j<=m)，可以将（c\*AiBj）mod p加入集合内。可以一直重复此步骤。求n个集合的并的大小。  1 ≤ n ≤ 10^4,1 ≤ m ≤ 10^5,1 ≤ Bi≤ 10^9,2 ≤ p ≤ 10^9且p为质数,1 ≤ Ai< p。 | 第i个集合的元素可以表示成Ai^(ΣBj\*cj){cj>=0}。设p的元根为g,令g^ri=ai，则元素可表示成：  g^(ri\*ΣBj\*cj)=  g^(ri\*gcd(B1,B2,…,Bm))=  不妨设t=gcd(B1,…,Bm)  g^(ri\*t)(mod p)  =g^(gcd(ri\*t,p-1)\*k)(mod p){k=0,1,2,…}  由于  gcd(a\*x,b)=gcd(gcd(a,b)\*x,b)  所以ri可以替换为gcd(ri,p-1)  又ri=(p-1)/di\*k其中di是最小的数满足a^di=1(mod p)，k=1,2,3…且(k,di)=1，所以ri=(p-1)/di。然后就是求p-1内能被ri(1<=i<=n)整除的数的个数。可以用dp来做。 | 时间  O(N\*sqrt(P)+sqrt(P)^2)  空间  O(N+sqrt(P)) |
| CF 303E | Random Ranking | N个人，第i个人得分为[li,ri]内随机一个实数。求第i个人得第j名的概率(1<=i,j,<=n)。  1<=n<=80，0<=Li<Ri<=10^9 | 要计算一个人排名为i的概率，可以算出j个人得分<li，k个人得分在[li,ri]，n-j-k-1得分>ri的概率p，然后i排名为j+1到j+1+k的概率都是一样的，枚举所有j,k即可。然后概率可以用dp来算。先枚举i，表示算i的排名，再枚举[li,ri]中的区间（离散化过的），F[l][j][k]代表前l个人（不包括i）有j个在这个区间之前，k个在之间的概率。 | 时间  O(N^5)  空间  O(N^2) |
| USACO 2013 US Open, Gold | Photo | 有N头奶牛排成一排，标号为1..N。小明拍了M张照片，照片i包含了从a\_i到b\_i的奶牛，每张照片中恰有一头奶牛有斑点。问至多有多少头奶牛有斑点。不可行输出-1.  n≤2\*10^5,m≤10^5 | Dp  F[i]代表前i个位置且i有牛的答案。  有条件可以得到转移的左右区间，然后用单调队列即可。  得到左右区间可以用单调队列，也可以用set。 | 时间  O(N\*logN)  空间  O(N) |
| CF 240E | Road Repairs | 给出n个点m条边的有向图，某些边费用为0，某些边费用为1，求最小树形图。并输出方案（费用为0的边不输）。无解输出-1.  N,M≤100000 | 最小树形图模板题。  用朱流算法。 | 时间  O(NM)  空间  O(N+M) |
| USACO 2005 December Gold | Cow Patterns | 给定长度为k的模式串和长度为n的母串，其中字符集大小为S。求模式串在母串中匹配成功的每个起点。匹配成功的定义为两个串的字符离散化后相同。例如 1,4,4,3,2,1和 2,10,10,7,3,2  1<=N<=100,000 ，1<=K<=25,000 ， 1<=S<=25 | Kmp  匹配成功的定义其实等价于前i-1个字符中比第i个字符小的和大的个数都相等。然后就是裸的kmp了。 | 时间  O(N\*S+K\*S)  空间  O(N\*S+K\*S) |
| CF 354D | Transferring Pyramid | 有一个n行的金字塔，第i行有i个格子。再给定k个格子，表示需要被修改。现在你可以用两种操作，一种是t i v，代表修改i位置的值为v，代价为3。另一种是t i v[1] v[2] … v[s]，代表修改以i位置为根的子金字塔的值为v[1],v[2]…v[s]。代价为2+s。求最小代价覆盖到所有要修改的位置。注意不需被修改的位置也可被覆盖。  1<=n,k<=10^5 | Dp  F[i][j]代表最右下角边长为i的三角形挖掉左下角边长为j的三角形之后的覆盖这个区域里面特殊格子的最小代价。  我们要求f[n][0]。  由于这个图形中第一列只会选择一个位置进行操作2，于是我们枚举这个位置，可以做到O(N^3)。观察到这个转移可以逆着for优化掉一个N，此时需要O(N^2)。再观察到全部用操作1的代价为3\*n，于是运用操作2的三角形大小<6n，即边长<sqrt(6n)。于是复杂度降为O(Nsqrt(N)) | 时间  O(N\*sqrt(N))  空间  O(N) |
| CF 311C | Fetch the Treasure | 有h间房间，从左到右依次排开，其中有n个房间有钱，分别是a1,a2,…,an，分别藏有c1,c2,…,cn美元。一开始 Rainbow在房间1，每次只能往前走k个房间。接下来有m个操作。共有三种操作。  操作1是1 x，作用是让Rainbow能够往前走x(1 ≤ x ≤ n)步;  操作2是2 x y (0 ≤ y < c[x],1<=x<=n)作用是将第x个藏宝房的金钱减去y;  操作3是3，要求输出Rainbow能够到达的藏宝房中金钱的最大值，并取走它。如果有两个房间金钱一样则取藏宝房id中最小的那个(1<=id<=n)。如果无法到达任意藏宝房则输出0。  最多只会有 20 个1号操作。  1 ≤ ai ≤ h, 1 ≤ ci ≤ 10^9 , h, n, m, k ≥ 1  h <= 10^18, n <= 10^5, m <= 10^5, k <= 10^4 | 背包问题dp  由于操作1的x很大，但是k很小，所以我们以模k分类，f[i]代表走到模k余i的最小只(1<=f[i]<=h)。  然后f[i]+k,f[i]+2k…这些都可以到达。  每加一个操作1，我们就用spfa的方式更新一遍f[]数组。然后扫描一遍藏宝房看有没有新增的课到达的房间。这步复杂度为O(K\*20+N\*20)  然后我们用数据结构（例如堆，平衡树）时刻维护能够到达的藏宝房的最大值。这样就可以解决此题了。 | 时间  O(N\*logN+K\*20+N\*20)  空间  O(N) |
| CF 295D | Greg and Caves | 给定n\*m的网格。现在定义网格的合法性：  能够找到L,R(L<=R)满足在L行到R行每行都恰有两个格子被染黑，并且其他行都是白的。并且两个黑格子差是先递增再递减，而且前一半每行的两个黑格子组成的区间都会被后一行的包含（相等也算），后一半相反。  求合法的方案数模1000000007后的结果。  1 ≤ N, M ≤ 2000 | Dp  F[i][j]代表满足前一半的条件下第i行两个黑格子位置差为j的方案数。  然后枚举差最大且最下面的那行i，再枚举黑格子位置之差j，则此时答案为f[i][j] \* (m - j) \* (f[n - i + 1][j] - f[n - i][j])。 | 时间  O(N\*M)  空间  O(N\*M) |
| CF 256D | Liars and Serge | 有n个人，每个人要么一直说真话，要么一直说谎。每个人都了解其他人的底细。现在问他们有几个人总是说真话，说真话的人会说出正确答案，说谎的人会说1到n中不是正确答案的任意一个。现在通过回答知道他们之中有k个人说谎。求有几种这样的回答。  1<=k<=n<=2^8,保证n是2的整数次幂。 | Dp  如果i的出现次数不是i，则说i的都说谎。  F[i][j][k]代表放置回答为小于等于i，已经放了j个数，并且已经知道有k个说谎的。  转移就是枚举放几个i。  这样是N^4的。但是由于n是2的整数次幂，所以打表就好了。 | 时间  O(1)  空间  O(1) |