|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试题编号 | 名称 | 题目大意 | 算法讨论 | 时空复杂度 |
| Codechef MAY 13 | Chef Protection Plan | 有T(T <= 100)组数据，每组数据输入一个格式串，再输入Q(Q <= 1000)个匹配串，问每个匹配串是否满足格式串。 | 先将待处理字符串的括号两边加上空格，保证每个语段的都可以被空格分开。然后对字符串进行语法分析，这里使用简化的递归递减分析的方法，这种方法不需要回溯，分析字符串的时间复杂度为O(N)。在分析的过程中，将其转化为正则表达式，即可转化为正则表达式的匹配，然后可以直接使用库文件<regex.h>来解决正则表达式的匹配问题。 | 时O(N)  空O(N) |
| Codechef MAY 13 | Queries on tree again | N(N <= 100000)个点N条边的无向连通图，Q次操作，每次操作对u到v的最短路径上所有边的权值取相反数，或求u到v的最短路径上连续边权之和的最大值。 | 首先将整个图分解成一个环和若干外向树。对每棵树进行树链剖分。在树链剖分后的每个路径上，用线段树维护每个线段上最大的连续区间和，包括左端点的连续区间和以及包括右端点的连续区间和。这样在线段树上便可以合并。区间翻转时我们需要在线段树上打一个标记，然后用lazy tag线段树懒惰下传。再用一个线段树维护环上的情况即可。单次操作时间复杂度O(logN)。 | 时O(QlogN)  空O(NlogN) |
| Codechef SEP 12 | Knight Moving | 骑士从(0, 0)开始，可以走到(u+Ax,v+Ay)或者(u+Bx,v+By)，有K(K <= 15)个障碍格，问走到(X, Y)有多少种走法。 | 如果A和B线性无关，则先忽略所有障碍点计算方案数，然后用容斥原理减去经过若干障碍点的方案数。否则我们对所有的坐标缩放gcd(A, B)使其线性无关。 | 时O(2KK)  空O(2KK) |
| Codechef SEP 12 | Annual Parade | 有N(N <= 250)座城市，M(M <=30000)条单向道路相连。选择一些路径，使路径的边权之和减去没有经过的城市数\*C最小。 | 先求任意两点之间的最短路径，然后建出费用流的图，求出带参量最小费用最大流，然后每个询问查询即可。 | 时O(N3logN)  空O(N3) |
| Codechef OCT 13 | Edit Distance on Grid | N行M列(N, M <= 100)的格子，可以交换相邻格子颜色或者染黑或者染白，使得每个格子都在一个四连通块中，求最小代价。 | 进行最优连通块覆盖，覆盖代价为连通块的代价之和。然后随机对连通块进行组合，使得组合的代价尽量小。然后枚举任意一对格子，对它们的连通块进行通路覆盖。 | 时O(N2 M2)  空O(NM) |
| Codechef NOV 14 | Chef and Churu | 数组有N个元素(N <= 100000)，有N个函数，每个函数返回一段区间的和。每次操作修改数组的一个元素或者询问若干函数的和。 | 一开始可以轻松的求得所有函数的返回值。对数组分块然后每次修改的时候可以用根号N的时间修改块的返回值。询问时对于部分块的和可以用树状数组维护。 | 时O(QlogN)  空O(NM) |
| Codechef APR 12 | Find a special connected block | N\*M的矩阵(N, M <= 15)，找到一个包含至少K(K <= 7)个不同数的代价和最小的连通块。 | 随机一些1~K的映射，最优解一定是树形的，所以可以进行dp。f[i][j][s]表示到达格子(i, j)时已经选中了s中的数。可以证明最优解可以在期望K!/K2次随机后得到。 | 时O(N2M2K!/K2)  空O(N2 M2K) |
| Codechef JAN 14 | Counting The Important Pairs | 不超过300000个点和边的简单无向图，求删掉两条边后图仍连通的方案数。 | 首先考虑一棵搜索树，那么每条边可以分为树边或非树边，删除非树边是不会影响连通性的。对整个图进行DFS标号，每条边的标号取决于连接它的非树边的集合，对它进行hash判重即可。 | 时O(N+M)  空O(N+M) |
| Codechef JAN 14 | Counting D-sets | 求N(N <= 1000)维空间中的整点的点集直径为D的等价类的个数。 | 求出直径<=D的等价类的个数，可以用f(D)-f(D-1)得到答案。用容斥原理刨去至少K维中直径没有到达D的方案数C(N, k) \* 2^(D^k \* (D + 1)^(N-k))。预处理组合数即可。 | 时O(N2)  空O(N2) |
| Codechef APR 12 | Substrings on a Tree | 问N(N <= 250000)个节点的字典树不同子串的个数以及Q(Q <= 50000)次查询第K小给定大小关系的子串。 | 先对字典树构建后缀自动机，然后便可dp求得不同路径个数即不同子串计数。由于大小关系随机生成，所以期望在不会匹配太长即可确定，沿着路径遍历即可。 | 时O(N+Q)  空O(N) |
| Codechef APR 12 | Similar Graphs | 对两个N(N <= 75)个节点的图进行重新编号，使得重合的边的数量最多。 | 不停地随机重新标号选出重合边数最多的方案输出。 | 时O(kN2)  空O(N2) |
| Codechef FEB 14 | Graph Challenge | 给一个N(N <= 200000)点M边的有向图，每次询问经过一个点的中间结点dfs编号均大于端点dfs编号的路径的条数。 | 先求出dfs编号，所求路径一定是一定包含dfs树上的非树边，可以求出询问节点连接的所有可达的非树边，以及所有反向可达的路径条数，组合相乘。 | 时O(N+M+Q)  空O(N+M) |
| Codechef FEB 14 | Count on a Treap | 维护一颗支持插入、删除、查询指定节点距离的treap。共N(N <= 200000)次操作。 | 关键在于维护节点的距离。需要快速求出其最近公共祖先的深度，可以通过动态维护倍增数组实现。如果删除造成的子链上移的相对位置不变，必要时进行重构即可。 | 时O(Nlog2N)  空O(Nlog2N) |
| Codechef DEC 14 | Kali and Devtas | 返回一个N(N <= 400)个点的生成树，使得生成树中被个数最多的直径最大的点圆覆盖住的点尽量小。 | 综合使用贪心和最小生成树的算法，在保证路径尽量直的情况下求出最小生成树。可以随机一个系数对路径夹角以及路径长度加权。 | 时O(N2logN)  空O(N2) |
| Codechef DEC 14 | Divide or die | 给一个N(N <= 360)度角，问能否尺规作图将其N等分，并构造一种方案。 | 若N被3整除，输出不可能。否则构造一个36度角和一个30度角然后相减再平分得到3度角，便可构造任何N不被3整除的情况。 | 时O(N)  空O(N) |
| Codechef NOV 13 | Queries With Points | N(N <= 100000)个简单多边形，在线回答Q次询问，每次回答包含一个特定点的多边形或者没有被任何多边形包含。 | 先检查一个点是否与多边形的轮廓重合。否则考虑这个点所在的与x轴垂直的直线上的所有边，可以通过这些边来判定它是否在每个多边形上。为了可以在线查询，需要维护一个函数式平衡树。 | 时O(NlogN)  空O(NlogN) |
| Codechef NOV 13 | Gangsters of Treeland | N(N <= 100000)个点的树，每次覆盖从根出发的一条路径或者询问某个子树中所有点到根的代价的平均值。 | 用DFS序对整个树进行树链剖分，覆盖操作可以转化为每条链上的区间加操作，因为每个帮会最多只会进入一次，贡献一条边的权值。子树平均值转化为区间求和。 | 时O(Nlog2N) 空O(NlogN) |
| Codechef DEC 11 | Short II | 给定质数p(p <= 1012)问有多少对数(a, b)满足ab被(a-p)(b-p)整除。 | 可转化为p(a + b + p)整除ab。共3种情况，p整除a和b，那么这样的(a, b)一定满足要求；p整除a但不整除b，对应着两种情况；p既不整除a也不整除b，那么a+b+p一定整除ab，恰好对应着一对(a, b)满足要求。 | 时O()  空O(1) |
| Codechef DEC 11 | Hypertrees | 求N(N <= 17)个点的本质不同的每条边连着3个点的树的计数。 | 状态压缩动态规划。将每个点当前的度数压缩成一维状态，然后连通性取决于度数的组合。所以每次加入一条边的时候连通性都不会受到非特征性状态的影响，枚举连边情况即可。 | 时O(3NN4)  空O(3NN) |
| Codechef APR 15 | Little Party | 有M(M <= 1000)个集合，共N(N<= 5)个元素，找到长度最小的基子集。 | 递归枚举每个元素是否被包含，找到包含这些元素的基子集的所有状态，求出满足每个状态的基子集的长度的最小值。 | 时O(3NM)  空O(M) |
| Codechef NOV 14 | The Spelling Problem | 有N(N <= 10000)个程序和两台电脑，每个程序在每台电脑上运行花费不同的时间，问运行所有程序的最短时间。 | 即要选出一些数使它们的和最接近总和的一半。可以先随机选数知道最接近总和的一半，然后求出剩下的数中删一个加一个最接近总和的一半的最小值。每次随机会以1/2的期望逼近最优解。 | 时O(N)  空O(N) |
| Codechef MAR 15 | Counting on a tree | N(N <= 100000)个节点的树，Q(Q <= 100)次修改，每次修改一条边的权之后询问有多少条路径上的边的最大公约数为1。 | 对树进行点分治，用容斥原理和莫比乌斯反演可以转化所求值为最大公约数为每个可能的约数时的路径条数。用并查集依次合并这些路径。每次询问影响的是连接这条边的所有路径，也可以用并查集维护。 | 时O(2c(N+Q2))  空O(N) |
| Codechef MAR 15 | Random Number Generator | 给一个K(K <= 30000)项的线性递推式，求数列第N项。 | 每一项可以表示为某K项的线性组合，两个相邻区间的特征可以合并，类似于快速幂的方法快速合并即可。合并时平方算法会超时，可以使用快速傅里叶变换进行优化。 | 时O(KlogKlogN)  空O(K) |
| Codechef JUN 11 | Attack of the Clones | 每个N(N <= 100)项的01串映射到0或1是布尔函数，有Q(Q <= 100)次询问，每次询问0-保留函数、1-保留函数、仿射函数以及自对偶函数集合的某种集合运算下对应着多少种布尔函数。 | 只需要计算16种情况下的布尔函数个数，即这种4种集合种其中所有子集对应的布尔函数集合的交集。前两种集合使得方案数减少一半。自对偶函数使得方案数开根号，仿射函数使得方案数变为2n+1，同理可处理出任意交集。 | 时O(Q)  空O(Q) |
| Codechef JUN 11 | Minesweeper Reversed | 有一个R行C列(1 <= R, C <= 50)扫雷地图，每次可以关闭一个格子，同时会关闭所有与它可能同时打开的格子，问最少关闭多少次。 | 关闭每个边界格子最多只会关闭旁边的两个边界格子。类似于集合覆盖问题，为边界格子的关系建出一张图，然后求最大匹配即可。 | 时O(NMlog(N+M)  空O(NM) |
| Codechef DEC 12 | Different Trips | N(N <= 100000)个节点的树，度数相同的点是相似的，问有多少条向根延伸的不相似的路径。 | 按每个点的度数建立树上的后缀数组，按height数组统计即可。 | 时O(Nlog2N) 空O(N) |
| Codechef DEC 12 | Quasi-Polynomial Sum | 给定一个D(D <= 20000)次多项式P在0~D-1处的值，求P(i)i在0~N-1处的和模M。 | 对原多项式进行D次差分后一定为常函数。Q的幂经过差分后成为Q-1的若干次幂。预处理组合数然后进行累加求值即可。 | 时O((DlogD+logN) logM)  空O(D+logN) |
| Codechef  FEB 15 | Devu and Locks | 求有多少N(N <= 109)位十进制数是P的倍数且每位之和等于1~M，允许前导0，答案对998244353取模。 | 按数位进行动态规划，每一位有PM个状态，且仅与上一位相关，可表示为矩阵。任意一个区间也可以表示成PM个状态，且相邻区间可以合并状态，用快速幂的方法可以将所有区间合并。合并时需要用快速傅里叶变换优化。 | 时O(PMlogPM logN)  空O(PM) |
| Codechef  FEB 15 | Payton numbers | T(T <= 10000)组数据，问一个三元组能否被分解成两个单位三元组的乘积。 | 可以建立一个三元组乘法到整数乘法的映射，然后将原问题转化为判定指数的问题，且也满足二次剩余性质。可以用Miller-Rabin算法检验满足质数性质的三元组。 | 时O(T)  空O(1) |
| Codechef  NOV 14 | The Spelling Problem | 给一段文本包含D(D <= 107)个字母和一个词库，一个词可能会出现多打、少打、打错一个字母或者交换两个字母的错误，输出改正后的文本使得错误尽量少。 | 用任何可能出现的错误改正每个不在词库中的词，用hash表查找是否在词库中，如果存在即纠正，否则继续匹配替换选项。 | 时O(D)  空O(D) |
| Codechef  DEC 13 | Query on a tree VI | N(N <= 100000)个节点的黑白染色的树，每次可以反转一个节点的颜色或者询问这个节点所在的连通块。 | 首先将其树链剖分。对于每次修改，找到它向上到达的第一个不同色的点，路径上按树链剖分分成很多的同色块，将这些区间同时加上这个点下方的同色的个数。 | 时O(Nlog2N)  空O(NlogN) |
| Codechef  DEC 13 | Petya and Sequence | 给一个N(N <= 30000)项轮换矩阵，问它是否是满秩的。 | 等价于检查是否存在n的约数d满足phid(x)整除原多项式。可以用快速傅里叶变换做多项式除法判断整除，然后逐个枚举判断。 | 时O(NlogN)  空O(N) |
| Codechef  JUN 15 | Connect Points | 给出连着N(N <= 70000)个点的M(M <= 300000)条边，问能否排成一个平面图且多任何一条边都会与现有的边相交。 | 先判断给出的图是否是平面图，然后再判断加入任意一条边是否都会变成非平面图。平面图可以用DMP算法线性判定。 | 时O(M)  空O(M) |
| Codechef  JUN 15 | Chefbook | N\*N(N <= 40)的格子上有M个格子上有数字，将每行加一个值，每列加一个值，使得每个数字都在给定的范围内，输出一组使得所有数字之和最小的可行解。 | 先将所求值转化为线性规划标准型。如果题目有解，说明不等式组 AP ≤ B有解，所以系数矩阵A是个非奇异矩阵。矩阵A的每个元素都是0或者±1容易推知det(A) = ±1。即最优解一定恰好在整点取到2。矩阵A每行恰好有一个1和一个−1。而我们知道如果是每列有一个1和一个-1的话，就是一个最小费用最大流问题了。转化为对偶问题即可。 | 时O(NM2) 空O(NM) |
| Codechef  FEB 12 | Find a Subsequence | 给一个N(N <= 1000)个数的数组和5位字符串表示表示相对大小关系，找出5个满足相对大小关系的子序列。 | 由于我们只关心数的大小，先将数列离散化。预处理出二维前缀和，即可在O(1)时间内回答是否存在数值在某一区间中的位置。枚举处于第2个和第4个的两个位置，然后用贪心的方法来求解。 | 时O(N2)  空O(N2) |
| Codechef  FEB 12 | Flight Distance | 给一个N(N <= 10)个点M(M <= 45)条边的无向图。要使每条边的权值等于连接这两个点的最短路长度，问最少改变的权值之和。 | 对于题目条件，可以很容易的列出一些线性规划的不等式，对于每条边拆成两个变量分别表示增量和减量，找出任意一组初始可行解即可使用单纯形算法求解。 | 时O(M4)  空O(M2) |
| Codechef  AUG 14 | Team Sigma and Fibonacci | 求满足x+y+z=N(N <= 1018)对M(M <= 100000)取模的值。 | 将式子中的一个未知数与另一个未知数交换，比如把以前的x都换成y，同时把以前的y都换成x，式子的值不变。fib数列模M的循环节长度是O(M)的，可以枚举余数进行计算。 | 时O(M)  空O(M) |
| Codechef  AUG 14 | Push the Flow! | 一个N(N <= 100000)个点的无向连通图，每个点最多属于一个简单环。Q(Q <= 200000)次操作，每次修改一条边的容量或查询S到T的最大流量。 | 先求双连通分量。最大流可转化为最小割。一类方案，删去一条连接两个环的边，例如图中红色的边。另一类方案，删去一个环内部的边，使这个环不再连通，从而使s、t不连通。用线段树来快速查询最小割的值。 | 时O(Qlog2N)  空O(N+M) |
| Codechef MAY 12 | Selling Tickets | 有N(N <= 200)道菜肴和M(M <= 500)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。 | 任务是找到最小的满足边数大于点数的边集，则答案是边数减一。这个子图有3种可能：两个度数为3的点之间有三条不相交的路径；两个不相交的环之间有一条简单路径连接；两个环在一个顶点处相交。枚举中间点，建树然后枚举每条变求出环的长度。 | 时O(N3+N2M)  空O(N2) |
| Codechef MAY 12 | Little Elephant and Boxes | 有N(N <= 30)个盒子，有一定概率获得一个钻石或者一定量的钱，有M(M <= 30)个物品，一定会购买尽量多的物品，问期望购买的物品数。 | 首先求出cost[i][j]表示购买恰好i个物品使用不超过j个钻石需要的最小金钱。将N个盒子分为两部分，暴力枚举前一部分的状态，二分出另一部分中满足要求的区间，求出区间中的概率之和以及对答案的贡献。 | 时O(MN22N/3)  空O(N2) |
| Codechef MAY 12 | Killing Gs | 有N只虫子和M个种杀虫剂(N, M <= 200)，每种杀虫剂可以使每只虫子以一定概率死亡，求最少花费多少代价购买杀虫剂使每只虫子都有超过90%的概率死亡。 | 贪心地每次选择贡献比花费最高的杀虫剂，直到满足要求为止。结合模拟退火算法，每次确定一些可选的杀虫剂，然后按贪心的算法求出每种状态的价值。 | 时O(TNM)  空O(NM) |
| Codechef  JAN 13 | A New Door | 在一个W\*H的矩形中画N(N <= 1000)个圆，问没有被画的区域的周长。 | 对每个圆计算其在矩形内且未被其它圆覆盖的长度并求和。将这个圆用弧度区间表示，枚举其它所有的圆并计算它们的重合部分覆盖住的弧度区间。对所有圆求和再加上矩形边框的未被覆盖住的长度。 | 时O(N2)  空O(N) |
| Codechef  JAN 13 | Cucumber Boy and Cucumber Girl | 有B(B <= 8000)个N\*N(N <= 60)的矩阵。计算有多少个矩阵对满足它们的乘积矩阵中至少有一个奇数的排列的个数是奇数。 | 满足要求的矩阵对的充要条件是它们的乘积的行列式是奇数。枚举其中一个矩阵，判断其它所有矩阵与它的乘积的行列式是否为奇数，可以用Gauss-Jordan算法快速求出。 | 时O(N2B+B2)  空O(N2B) |
| Codechef  AUG 11 | Shortest Circuit Evaluation | 给一个长度不超过30000的布尔表达式和每个变量为true的概率，调整表达式的顺序使期望的计算次数最少。 | 只能调换and连接的变量和or连接的变量。将and连接的变量按期望的计算次数除以这个变量为false的概率从小到大排序即可。or同理。 | 时O(N)  空O(N) |
| Codechef  AUG 11 | Something About Divisors | 对于给定的正整数B和X(B <= 1012, X <= 60)，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 <= B)能整除N\*X。 | N会被不同的K重复计算，所以我们需要统计的是对于K满足K|NX但不存在J|NX的N的个数S(K)，即K是最大的能整除NX的N的个数，答案为。用容斥原理转化式子，再优化枚举即可。 | 时O()  空O(X) |
| Codechef  AUG 15 | Future of draughts | 有T(T <= 50)个图，每张图N(N <= 50)个点。每次询问一个连续区间中的图，在每张图中任意选一个起点，每次将若干个点走一步，问最后都回到起点的方案数。 | 对于每一个无向图G，令 H = G + I，计算出它的tr[Hi](0 <= i <= k)，可以使用大步小步算法或特征多项式算法。然后计算出P(i, j, k)，再用FFT优化卷积计算出ans(i, j, k)。 | 时O(T2KlogK)  空O(TNK) |
| Codechef  AUG 15 | Simple Queries | 有N(N <= 100000)个数和Q(Q <= 100000)次询问。查询所有三元组的乘积之和、删除、插入、修改以及查询区间不同的数的个数。 | 将整个平面分为块，分别处理块内信息与块外信息。每次询问一个二维区间时，包含的有若干个整块与若干个不完整的部分。对于整块，必然是一个矩形区域，所以维护一个二维树状数组能够方便地回答区间总和；对于不完整的部分，由于每一行或每一列最多同时存在一个点，直接枚举该部分的点就足够了。 | 时O(N)  空O(N) |
| Codechef  MAY 14 | Dynamic Trees and Queries | N(N <= 50000)个点的树，M(M <= 50000)次操作，增加一个叶节点，子树加、删除子树、询问子树和。 | 预处理出括号序列。之后维护每个操作。操作1新建一对括号对应这个节点，并将其插入点key的左括号右侧，其它操作可以对括号序列进行区间操作，可以用平衡树维护。由于括号序列上一个点会被算两次，输出答案的时候要除以2。 | 时O(Nlog­N)  空O(N) |
| Codechef  MAY 14 | Sereja and Subsegment Increasings | 有两个长度为N(N <= 100000)的数列，可以将第一个数列的某个区间中的数加一再模4，问最少多少次操作可转化为另一个数列。 | 对于非腐向，如果我们给它增加4，最多只能再给一项减少4，这显然不会更优。对于非正项，如果我们给他减少4，答案不会改变，显然也不会更优。所以问题转化成在第i项之前有di/4项可以选择，最大化贡献，从大到小贪心即可。 | 时O(N)  空O(N) |
| Codechef  MAY 14 | The Malaysian Flight Search | 一张地图由N(N <= 5000)个边界点描述。有M(M <= 50)种飞机，每种飞机有费用和搜索范围。每次可以将一个飞机派到(x, y)，需要付出C \* (x + y)的费用。要确定关键点的位置并最小化费用。 | 我们每次可以从所有点中随机一个点，之后枚举每一种飞机，计算出每一架飞机飞向该点的估价函数。之后只要选择一个估价函数最小的飞机即可。不断重复以上过程直到只剩下一个点，再派一个R为0的飞机探索该点就好了。 | 时O(kM)  空O(M) |
| Codechef JUL 11 | Billboards | 在公路上有N(N <= 109)个广告牌，大厨会选择一部分广告牌放广告。大厨希望任意连续的M(M <= 50)个广告牌中至少有K(K <= 50)个餐馆广告。问使用最少的广告牌的方案数。 | 如果M整除N，可以证明需要最少K\*(N/M)个广告牌。考虑M不整除N的情况，如果N % M <= M − K，那么每一组的前N % M个位置都应该是0。如果N % M >= M − K，那么每一组最后M − N % M的位置都应该是1。同样可以使用杨氏矩阵解决。 | 时O(KMlogN)  空O(1) |
| Codechef JUL 11 | Trial of Doom | 房间有N\*M(N <= 40, M <= 109)的格子，每个格子是红色或者蓝色。从(1, 1)八连通走到(N, M)，只能走蓝色格子，走到一个格子使得它旁边的四个格子变色，问能否走到。 | 于(x, y)这个方格，如果它是红色的，那么可以操作(x − 1, y)来使它变成蓝色。这样每个方格都代表了第一列方格的状态，这可以用递推来计算，在同一行中会出现循环，可以通过循环节来计算y比较大的情况。红色方格的异或值就是需要达到的状态，变量就是最后一列的状态。这可以通过线性基或是高斯消元来解决。 | 时O(N2K)  空O(NK) |
| Codechef JUL 11 | Large Kitchen | 一个N\*M(N, M <= 100)的图，填入尽量多的黑色方格，使得黑色方格四连通且没有回路。 | 问题就是在方格图种放入一棵树，使得树的节点最多。Border表示在边界上的白格个数，Corner表示在角落的白格个数，Neigh 表示相邻白格对数。为了让黑色格子最多，就需要让Border + 2Corner + Neigh最小。构造出一组较优的解，并根据N, M的情况在局部进行微调。 | 时O(NM)  空O(NM) |
| Codechef APR 14 | Chef and Tree Game | 有一棵N(N <= 105)个节点的树，树上每一条边的颜色都为红色或者蓝色。现在两个人轮流进行操作，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，删除后和树根不连通的部分将被删除，若干轮之后不能操作的人算输。问谁能赢。 | 设当前局面的权值为V，在第一个人操作之后局面权值的最大值为R，第二个人操作之后局面权值的最小值为B，那么有V = {R|B}。可以发现对于根的每一个孩子来说，游戏都是完全独立的，所以我们可以把问题简化成根只有一个孩子的情况。设G的权值为x，当根和G连接的边为红边时，令p为最小的正整数使得x + p > 1，那么当前局面的权值为(x + p) / (2p - 1)；当根和G连接的边为蓝边时，令p为最小的正整数使得x − p < −1，那么当前局面的权值为(x - p) / (2p - 1)。再用高精度运算即可。 | 时O(Nlog2N)  空O(NlogN) |
| Codechef APR 14 | Make It Zero 3 | 长度为N(N <= 1015)的数组按指定随机方法生成，每次可以选定一个区间将其中的数加一个值模P(P <= 10)，问最少多少次操作把所有数变成0。 | 先进行差分，然后等价于把数组分成若干个和为0的子序列且每一个元素恰好出现了一次，最大化分成的子序列个数。可以使用类似单纯形的方法，先随便求出一个解，然后逐步逼近最优解。可以先用dp预处理所有分裂方法，然后选择最简分裂方法进行分裂。 | 时O(MP5)  空O(P4) |
| Codechef SEP 13 | Two Roads | 给定平面上的N(N <= 100)个点，求出两条直线，使得所有点到这两条直线的距离的较小值的平方的平均值最小。只需要输出这个最小的平均值。 | 显然我们只需要求出这些距离的平方和的最小值。按照极角序枚举第一条划分直线，则在极角增大时离散化的结果会有一些点对交换。总共的点对交换次数为n(n-1)/2。可以通过点对交换维护离散化的结果。 | 时O(N3)  空O(N) |
| Codechef SEP 13 | To Queue or not to Queue | 给定一个初始为空的字符串，要求支持在最后添加字符和删除第一个字符，Q(Q <= 100000)次操作，每次操作后询问本质不同的子串个数。 | 如果构造出字符串的后缀树，则本质不同的子串个数为后缀树的所有边的长度之和。使用后缀树的在线构造算法（Ukkonen算法），可以在最后添加字符时维护后缀树。现在考虑删除第一个字符的操作。这个操作相当于在后缀树中删除最  长的后缀。这个后缀一定对应后缀树中的一个叶子结点。在后缀树中删除一个后缀相当于在 Trie 树中删除一个字符串。只需要删除这个串对应的结点，然后如果之前这个结点的父亲结点变为叶子结点则需要继续用同样方式删除，直到父亲结点不是叶子结点为止。即需要删除所有只属于这个串的结点。 | 时O(Q)  空O(Q) |
| Codechef NOV 12 | Arithmetic Progressions | 长度为N(N <= 100000)的数列，问有多少个等差三元对。 | 先分块，有三种情况：三个数都在当前块；两个数在当前块，另一个数在前面的块或后面的块；一个数在当前块，一个数在前面的块，一个数在后面的块，这种情况需要用快速傅里叶变换统计。 | 时O(N)  空O(N) |
| Codechef NOV 12 | Martial Arts | 一个点数为N(N <= 100)的二分图，找到一个完全匹配，使得去掉一条边后的第一种权值之和与第二种权值之和的差值最小。 | 枚举对手删哪一条边。按顺序枚举删的边，只需要维护两种操作，加边和强制匹配这一条边，查询最大匹配。动态地维护KM算法中的顶标即可。 | 时O(N4)  空O(N2) |
| Codechef NOV 12 | Many Left | N\*N(N <= 30)的棋盘，有些棋子，类似于跳棋，相邻的可以跳过去，被跳过的就消失了。一直跳到再也跳不了，求尽量优的方案使剩下的棋子最多。 | 先随机一个可行解，然后随机调整其中的某些步骤，重构后面的部分，判断这个解是否更优，如果更优即采纳。在所给的时间内尽可能多地调整，找到一个尽可能优的解。 | 时O(kN2)  空O(N2) |
| Codechef OCT 12 | Max Circumference | 给一个三角形和N(N <= 500)个操作，每个操作可以使三角形的同一个点位移(x, y)，从中选择最多K个操作使得三角形周长最长。 | 即找到合适的u, v，最大化线性函数f(X, Y) = uX + vY。假设已经确定了u, v，那么每个操作对f(X, Y)的贡献都是独立的，排序后取前K个正权操作即可。把向量按极角排序，在相邻两个向量夹角中随意选取一个向量作为(u, v)，排序计算。 | 时O(N2logN)  空O(N2) |
| Codechef OCT 12 | Ciel and password cracking | 有C(C <= 1000)个点，选择其中K(K <= 5)个破解密码，并通过分配电脑中心，选择电脑，调整连接方案，来最小化期望所需时间。 | 设Time[i][j]为第i个密码在第j个点破解的最小期望时间，如果这个已经完成了计算，只需要一个状态压缩动态规划便可以解决选择电脑中心的问题，所以主要问题是计算Time[i][j]。通过二分判断差值再用指数分布函数累加即可。 | 时O(2KCK)  空O(2KC) |
| Codechef OCT 12 | Maximum Sub-rectangle in Matrix | 有一个H\*W(W, H <= 300)的矩阵，选出一个和最大的可以不连续的子矩阵。 | 先随意生成一个权值为正的状态，然后不断调整，每次选出增益最大的一行或一列进行调整，直到调整到最佳方案。 | 时O(kWH)  空O(WH) |
| Codechef JAN 15 | Ranka | 两个人下围棋，构造一种走法使得，每次下完后棋盘上的局面之前没有出现过，要求N(N <= 10000)步。 | 先走满整个棋盘空一格然后把那一个填上，每次移动一个空格子来达到和之前不重复的效果即可。 | 时O(N)  空O(1) |
| Codechef JAN 15 | Xor Queries | 维护一个数组，共M(M <= 500000)次操作，在最后删数或加数，查询某一段中与指定数异或最大的数，或第k小的数。 | 维护一个二进制字典树，并在每个节点存储经过这个节点的所有数的编号。每次查询都从顶部向下走，动态更新查询的信息。 | 时O(MlogM)  空O(M) |
| Codechef OCT 14 | Children Trips | 给定一棵N(N <= 100000)个点的树，边权为1或2。每次给定点u和点v，并给出距离d。一开始在点u，每天从当前点出发向点v走不大于d的距离到达一个点，询问从点u走到点v至少需要多少天？ | 我们将u到v的路径分为两段：从u到lca和从v到lca。易证，方向  相反对答案并没有太大影响。对于每段路（以u到lca为例），我们从u一天一天往上走走到最接近lca的点，还没到lca，记该点为u，同理得出点v。那么该询问的答案为u到u的天数v到v的天数u到v的天数。 | 时O(N)  空O(NlogN) |
| Codechef OCT 14 | Union on Tree | 有一棵N(N <= 50000)个点的树，边长为单位长度。每个询问给出一些警卫，每个警卫在一个点保护一定距离内的点，求有多少个点受到保护。 | 每次询问，我们先对警卫的保护范围进行松弛操作。建出虚树，对于虚树的每一条边上的点，只可能是由边两端的警卫进行保护（因为进行了松弛操作） 。把虚树上的每条边都按分开处理的方法算一遍，可以分别用点分治和可持久化线段树。 | 时O(NlogN)  空O(N) |
| Codechef OCT 14 | Chef and Painting | 一个N\*M(N, M <= 100)的矩阵，一开始有K个格子被染了色。选择一个没染色的格子，选取左右或者上下两种方向的其中一种，一路将这个方向上没有染色的格子染色。 | 暴力贪心，从上到下每行从左到右找到一个没有染色的格子，然后以其为开头向左或向下画，取一个比较优的方向（能染更多的格子）染色，然后循环这个操作。多次随机取参数最优即可。 | 时O(kNM)  空O(NM) |
| Codechef NOV 11 | Luckydays | 给定一个模P(P <= 10007)的二阶线性递推数列和一个数C，Q(Q <= 20000)次询问每次询问一个区间中有多少个数等于C。 | 先求序列的周期，可以用大步小步算法。然后求出C后面跟每一位数的最小出现位置，也可以用大步小步算法。然后将所有上述的出现位置从小到大排序，一个区间可拆成两个区间相减。可将统计任务拆成第一个循环之前和第一次出现之后，都可以用二分查找统计，相加即可。 | 时O(P1.5+QlogP)  空O(P1.5) |
| Codechef NOV 11 | Colored Domino Tilings and Cutsontest | 构造一个N\*M(N, M <= 500)的棋盘覆盖，使得棋盘的割尽量少，此前提下让染色数尽量少。 | 首先，排除N\*M是奇数的情况，这种情况一定无解，否则一定有解。N和M较小时可以暴力求解。N >= 5且M >= 6时一定可以没有割线。以通过一种方法轻松地将一个没有割的棋盘覆盖拓展成更大尺寸的没有割的棋盘覆盖。这样一定可以被三染色。染色方案搜索剪枝即可。 | 时O(NM)  空O(NM) |
| Codechef OCT 11 | The Baking Business | 某厨师出售一些例如饼干，面包等可以用不同尺寸包装的产品。他在不同省份的不同城市甚至城市进一步划分成的地区出售产品。他还记录了每一次购买的顾客的性别和年龄。S(S <= 100000)次询问，每次查询基于各种参数的销售报告。 | 这其实是一道模拟题。考虑到查询有参数缺省的话不能进行枚举复杂度可能过高，我们考虑在出售的时候就预处理出某个范围内的出售总数。注意要枚举产品的所有情况以及地点的所有情况进行组合。查询因为是区间和，但范围并不大，所以我们可以暴力枚举。 | 时O(S)  空O(1) |
| Codechef OCT 11 | Sine Partition Function | 求所有系数之和为N(N <= 109)的M(M <= 30)个整数倍X的正弦值之积的和。 | dp[i][j]表示N = i, M = j时的答案，dp[i][j] = dp[i-1][j-1]sinX + dp[i-1][j]2coxX - dp[i - 2][j]。再用矩阵快速幂进行优化即可。 | 时O(M3logN)  空O(M2) |
| Codechef APR 13 | String Query | 共Q(Q <= 150000)次操作，要求支持在串最前面、最中间以及最后面的插入删除操作，询问一个字符串在串中出现次数。 | 我们维护一个数据结构SL=L+R，L, R都为一棵后缀平衡树。这样的话，我们就可以维护最前面与最后面的动态插入删除了，当L或R为空时，把整个串均分即可。而至于要维护中间插入删除的话，我们再维护一个数据结构G=SL+SR。保证SL最后面一个一定是中间即可，即对于每次插入删除动态维护一下即可。对于询问，在串内就直接在平衡树中处理，在串间，就取出来做kmp匹配即可。 | 时O(Qlog2Q)  空O(QlogQ) |
| Codechef APR 13 | Inverse Binomial Coefficient | 求最小的K，使得C(2N-1, K)=R在模2N意义下成立，N <= 120。 | 我们先可以对其进行质因数分解，这样我们只需要预处理所有的质数就可以了（只有29个）。然后对于pb的求法。我们可以使用分组的方法，通过一些预处理，再用上高精度即可。 | 时O(N3)  空O(N) |
| Codechef APR 13 | Fault Tolerance | 有N(N <= 200)个变量，以及(M <= 1000)个异或方程，要求删除尽量小的方程，使得不能解出N个变量的唯一解。 | 查看存不存在一列可以被其他列线性表出的，如果有，那么没有唯一解，否则就有唯一解。不断进行随机调整，随机选择一列，然后再随机选择该列上的一个1，然后就把其他在该1所在行上有1的列进行异或找最优值，不断地重复操作。这样之后，每一列就不是初始的列，而是一些列的异或之后得到的结果了。可以用位运算优化。 | 时O(kNM)  空O(NM) |
| Codechef JUL 15 | A game on a graph | N(N <= 2000)个点的无向图，从一个点开始走，每人用最优策略移动到没有走过的点，不能走的算数，求胜利点个数。 | 胜利点的充要条件是不存在包含它的最大匹配。用带花树算法求出最大匹配。枚举每一个孤立点，仿照带花树寻找匹配的过程，从这个孤立点出发寻找增广路。每一条长度为偶数的增广路的终点，都可能在最大匹配中成为孤立点，也就是游戏中的胜利点。 | 时O(NM)  空O(N + M) |
| Codechef JUL 15 | Easy Exam | 将一个K(K <= 109)面骰子抛N(N <= 109)次，问抛出每个数字的次数的乘积的F(F <= 1000)次方的期望。 | 令dp[i][j]为当前已经考虑了前i个数字，它们对答案产生贡献时，需要占用j个时刻的方案数。dp[L][i]的值是一个生成函数L次幂的i次项系数。可以使用指数生成函数。再用快速幂和快速傅里叶变换优化卷积即可。 | 时O(F2 + LFlog(LF)logL)  空O(F2) |
| Codechef AUG 13 | Music & Lyrics | 求前面W(W <= 500)个字符串中的每个在所有N(N <= 100)个字符串中出现的次数和。 | 对所有搜索串构建一个AC自动机，然后用每个目标串跑一遍，在经过的节点上标记加一，然后用树形动态规划沿着fail指针求出每个节点上的访问次数，输出即可。 | 时O(N + W)  空O(W) |
| Codechef AUG 13 | Prime Distance On Tree | 问一棵N(N <= 50000)个点的树上两个点距离是质数的概率。 | 每次选择树的重心，在树上进行点分治，统计经过重心的每种距离的路径的条数。用快速傅里叶变换优化卷积即可，这里要注意将子树从小到大排序再进行卷积。 | 时O(Nlog2N)  空O(N) |
| Codechef AUG 13 | Deleting numbers | 有N(N <= 100000)个数，每次可以删除等距的相同的一些数，最优化删除所有数的步数。 | 从后往前，对于只出现一次的数直接删掉，否则找到一个和当前数相同的数一起删掉，并继续向前扩展，用数据结构维护即可。 | 时O(NlogN)  空O(N) |
| Codechef JUN 13 | Two k-Convex Polygons | 给定N(N <= 1000)个棍子的长度和整数K(K <= 10)，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。 | 若有超过70根棍子，则必定存在两个K边形，排序后暴力寻找两遍即可。首先排序后扫一遍，寻找第一组，之后删掉第一组寻找第二组，找到了输出方案。否则，说明K不超过70，可以枚举哪一段再枚举划分并判断。依旧没有找到输出No。 | 时O(NlogN)  空O(N) |
| Codechef JUN 13 | Count Special Matrices | 问这样的N(N <= 107)阶方阵的个数：沿对角线对称，A[x][y] <= max(A[x][z], A[z][y])，每个数都在1~N-2之间且每行每列都出现过1~N-2。 | 你要寻找的矩阵和距离矩阵非常类似（满足类似的三角不等式）。实际上，满足(a)-(c)的矩阵被称为超度量空间（可以理解为图的某种最短距离矩阵），而X还可以代表图的点集。答案等于N!(N-1)!/2N-1(N/2 - 2/3 - HN-1/3)，处理出逆元，然后计算即可。 | 时O(N)  空O(1) |
| Codechef MAR 12 | Evil Book | N(N <= 10)个厨师，打败一个需要花费一定代价并获得一定魔力，也可以花费X魔力使一个厨师的代价和魔力都除以3，问收集666点魔力的最小代价。 | 我们可以使用搜索。每次选择一个未被挑战的厨师，枚举其需要使用的帮助次数k，进入下一层搜索。用已经使用的帮助次数作记忆化。再加上可行性剪枝和最优性剪枝即可。 | 时O(N22N)  空O(N) |
| Codechef MAR 12 | Ciel and Earthquake | R(R <= 8)\*C(C <= 1018)的网格图，每条边有p的概率被破坏，问(1, 1)到(R, C)连通的概率。 | 可以用类似于插头dp的方法用最小表示方法表示连通性状态。每增加一列，其概率缩小的比值是收敛的，所以取一个足够大的列数计算即可。 | 时O(22RR3)  空O(2RR2) |
| Codechef JUL 12 | Dynamic GCD | N(N <= 50000)个点的树Q(Q <= 50000)次操作，维护查询路径最大公约数及路径加一个值。 | 序列的最大公约数就变成了自身和间隔的最大公约数的最大公约数。用动态树维护即可。 | 时O(QlogN)  空O(N) |
| Codechef JUL 12 | Equivalent Suffix Tries | 字符串长度不超过L(L <= 100000)求与给定字符串后缀字母树等价的个数。 | 首先找出后缀M和其他长后缀A的{lca(M,A)}-1的最大值H。（显然H<L，否则L可以更大）这样一来，st[N-L..N-L+H-1]便容易确定。然后用后缀数组（满足条件三）找出满足条件一的后缀所在的排名范围，再一个个比较排除掉条件二中不相等的字符。统计满足条件的位置个数。 | 时O(NlogN)  空O(N) |
| Codechef MAR 13 | Little Elephant and Colored Coins | 在N(N <= 30)种硬币中各选择一些（可以不选），使硬币面值之和为 S，在满足此条件基础上，希望硬币颜色种类尽量多。 | dp[j][i][color]表示已经考虑前j种硬币有color种颜色，满足之前条件的最小的x。分三种情况讨论，小于S表示能够构成值S且使用了c + 1种颜色，且一定使用了硬币M；等于S表示能够构成值S且使用了c种颜色，且一定没有使用硬币M；大于S表示不能够构成值S。处理这些情况对答案的贡献，即为我们的最终答案。 | 时O(NQ)  空O(NQ) |
| Codechef MAR 13 | Making Change | 有N(N <= 50)个整数，两两互质，且每个整数不超过500。每个整数可以选多次，也可以不选，问存在多少种选法使得所有数的和等于S。 | 求出生成函数G(x)，答案即为G(x)中xS项系数。因为所有的d是互质的，所以我们可以对原式进行部分分式分解。可以使用广义二项式定理将这个式子展开，最后使用扩展欧几里得算法即可。 | 时O(N4)  空O(N2) |
| Codechef AUG 12 | A Game of Thrones | 给定N(N <= 500)个数字，轮流取数，每次只能取与上一个之差一个质因子的数，问先手必胜时第一个选的数的最小值。 | 首先如果将每个数看作一个点，将两个满足相差为一个质因子的数之间连一条边。那么会发现构成的图是一个二分图。然后，发现对一个转化过去的二分图做最大匹配，先手选择任意一个最大匹配中的未配点都可以保证先手必胜。 | 时O(N2logN)  空O(N2) |
| Codechef AUG 12 | Two Magicians | N(N <= 7777)个点的无向图，轮流加边，也可以消耗一点法力值移动到不与对方在一个连通块的点，问是否先手必胜。 | 可以发现先手是否必胜其实与最终状态的边数的奇偶性有关，不妨将这个边数设为A。A减去已有的边数，如果为奇数则先手胜，否则后手胜。一种情况是一开始就只有两个联通块，且都为奇数，那么最终状态就固定了，可以直接得出答案。另一种当前总共的联通块为三个，且有两个为奇数，而先手希望最终状态为第一种情况，那么可以将其中一个奇数块与偶数块合并那么，先手仍然必胜。 | 时O(N)  空O(N) |
| Codechef  JUN 12 | Cool Numbers | T(T <= 100000)组数据，每次给定一个N(N <= 101000)询问比N大和比N小的第一个数，满足存在一到三个数位使得它的数字和减去这些数位的数字和S的S次方是它的倍数。 | 如果一个数只有1~3个数位上的数不是0，那么这个数明显是满足条件的。否则是第二类数，可以找出所有第二类数，并进行排序。设读入的数数位长度为a，则我们可以求出大于a的最小的第一类数和小于等于a的最大的第一类数，然后从第二类数中进行二分，求出大于a的最小的第二类数和小于等于a的最大的第二类数。最后再进行两次比较求出答案。 | 时O(TlogN)  空O(logN) |
| Codechef  JUN 12 | Expected Maximum Matching | 左边有N(N <= 5)个点右边有M(M <= 100)个点，给出每个左边的点和右边的点连通的概率，求期望最大匹配数。 | 根据hall定理，如果左边存在k个点和右边的大于等于k个点相邻，那么这个图一定存在一个大小为k的匹配。所以我们可以记录左边的每个点集是否满足条件然后每次枚举右边的点，进行一个状压dp，仅保留合法状态即可。 | 时O(2NN2M2)  空O(N2M) |
| Codechef  JUN 12 | Closet Points | 给定N(N <= 50000)个三维空间中的点，Q(Q <= 50000)次查询每次询问一个点在所有给定点中欧几里得距离最近的点。 | 将给定的点建出三维k-d树，然后每次在三维k-d树上查询。可能会超时，可以仿照暴力算法卡时，限制询问数量或时间，保证不超时。 | 时O(QlogN)  空O(N) |
| Codechef  OCT 13 | Three-Degree-Bounded Maximum Cost Subtree | N(N <= 100)个点的带权连通无向图，保证其中每个点双连通分量大小小于10，求一个3点度限制最大生成树的方案数。 | 对于两个点双连通分量，可以分开来考虑它们的3点度限制生成树，并且只需要记录下每个生成树中割点的度数就可以很容易地合并这两棵树。我们可以对于每一个连通分量单独进行状态压缩dp，然后对于不同的连通分量，需要记录的仅仅是它们的割点的2度数。由于整个图是连通图，所以所有的点双连通分量组成了一个树形结构，这一部分用树dp来处理即可。 | 时O(99N)  空O(N2) |
| Codechef  OCT 13 | Fibonacci Number | 求斐波那契数列中第一次出现模P(P <= 2\*109)余C的项数。 | 首先，由于P模10为质数，我们可以推断P在模5意义下余正负1，那么根据二次剩余的知识，在模P意义下一定存在5的平方剩余。所以，在此可以用二次剩余的知识直接求出根的n次方，接着用大步小步法求出n，判断是否符合假设即可。 | 时O()  空O() |
| Codechef  MAY 15 | Chef and Balanced Strings | N(N <= 100000)个字符的字符串，每次询问一个区间中所有字母出现次数均为偶数的子串的长度的零或一或二次方和。 | 考虑到此题要维护的数据比较复杂，难以用常见的高级数据结构来维护。但是分块这种简单策略在本题上有良好的表现。首先将序列分成根号N块。块内可以暴力计算，块间贡献通过预处理算出。 | 时O(N)  空O(N) |
| Codechef  MAY 15 | Counting on a directed graph | N(N <= 100000)个点M(M <= 500000)条边的有向图，统计无序对(X, Y)的个数，存在一条从点1到点X的路径，和一条从点1到点Y的路径，且两条路径除了点1以外没有公共点。 | 使用Lengauer-Tarjan算法计算支配树。在计算半必经点的过程中如果某个节点的前驱结点的时间戳比它大，需要考察这个前驱结点的所有时间戳比当前考虑节点大的祖先。计算最近必经点的过程中也要考虑它和半必经点之间的路径上的所有节点。我们可以按照时间戳逆序计算每个节点的半必经点，此时所有时间戳更小的节点仍未计算，在考察时间戳大的祖先时只要考察它在已计算节点森林中的所有祖先。维护已计算节点森林可以用并查集来实现。 | 时O((N+M)logN)  空O(N+M) |
| Codechef  MAR 14 | Chef and Graph Queries | 给定一个N个结点M条边的无向图(N, M <= 200000)。共 Q(Q <= 200000) 个询问，每次询问只考虑编号在l到r之间的边时，图中的连通块数。 | 按照r关键字对询问排序。 使用LCT来维护由编号1~r的边形成的最大生成森林。当我们发现新加入的边在同一个连通块内时，我们就可以将它们俩路径上编号最小的边删除，并用新边来替换。询问时我们查询当前LCT上权值大于等于l的边的数目，并设为C。这可以利用其它数据结构来完成，例如线段树或树状数组。 | 时O((Q+M)logN)  空O(N+M) |
| Codechef  MAR 14 | The Street | 维护两个N(N <= 109)元数组，M(M <= 300000)次操作，第一个数组与等差数列取最大，第二个数组与等差数列对应位相加，或求两个数组对应位之和。 | 操作之间不会互相影响，可以分开处理两个数组。对于第一个数组，两个等差数列最多只会有一个交点，所以用分成的若干块分别更新线段树打标记即可。对于第二个数组维护一个线段树即可，标记不用下传。 | 时O(Mlog2N)  空O(MlogN) |
| Codechef  FEB 13 | Room Corner | N\*N(N <= 2500)的地图上有一个直角多边形，每个角上有一个小朋友，相邻小朋友可以交换位置，Q(Q <= 10000)次询问两个小朋友相遇的最短时间。 | 假如我们把相邻的小朋友连一条边，小朋友就可以沿着这些边进行移动和交换位置。可以看出，所有小朋友最后会连成一个环。我们只要弄出这个环，并求出环上每一条边的长度（也就是移动所需的路程），后面的问题就会比较好办。显然这个环是沿着房间的边界的。我们任取一个小朋友，让他沿着墙逆时针绕着房间走一整圈，那么就相当于把环给遍历了一遍。每次询问用它们的中点二分寻找最近的点即可。 | 时O(N2+QlogN)  空O(N2) |
| Codechef  FEB 13 | Observing the Tree | N(N <= 105)个点的树，M(M <= 105)次操作，在线路径加等差数列，查询路径和或回到某个历史版本。 | 首先树链剖分，可转化为链上操作。用可持久化线段树维护链上的和，打一个等差数列标记即可。每当修改这个结点时，不在原来的结点上修改，而是将结点复制一份后在修改。可以使用标记永久化。 | 时O(Nlog2N)  空O(Nlog2N) |
| Codechef  SEP 14 | Rectangle Query | 在二维平面上，有Q(Q <= 105)次操作，插入一个矩形，删除一个矩形，询问当前有多少个矩形与询问矩形有公共点。 | 不难发现，我们关心的是4个偏序条件。将补集转化，可以发现我们最多只需要同时考虑2维的偏序(算上操作顺序则为3维)，可以通过分治套树状数组解决。 | 时O(Qlog2Q)  空O(Q) |
| Codechef  SEP 14 | Fibonacci Numbers on Tree | N(N <= 105)个点的树，M(M <= 105)次操作，在线路径加斐波那契数列，查询子树以及路径和，或回到某个历史版本。 | 首先按dfs序树链剖分。在链上维护若干棵可持久化线段树。这棵线段树需要快速地支持对一个区间内所有数从左往右或从右往左依次加上斐波那契数列从某项开始的连续若干项以及区间求和。对于线段树上每个区间，记录从左往右、从右往左的各一对和，以及区间内的总和即可。可以用斐波那契数列通项公式快速计算。 | 时O(Nlog2N)  空O(Nlog2N) |
| Codechef  SEP 11 | Short | 给定n(n <= 105)和k(k <= 1018)求出所有(a, b)使得ab - n可以被(a - n)(b - n)整除。 | 由于n的范围很小, 所以我们可以简单的枚举n。然后倘若我们枚举了c的值, 那么我们可以算出cn + n2 − n, 枚举其约数, 判断即可。可以证明c不超过3n。c较大时可以改为枚举约数。 | 时O(NlogK)  空O(N) |
| Codechef  SEP 11 | Counting Hexagons | 求边长不超过N(N <= 109)的合法的最长边至少为L的六边形个数。 | 首先枚举最大的一条边M。那么问题转化为了求5个单调不减的数, 使得和大于M, 最大的数小于M, 且相同数的个数小于K。用数位dp，压位并预处理一些转移方案数即可。 | 时O(log7N)  空O(log6N) |
| Codechef  JUN 14 | Two Companies | N(N <= 105)个节点的树，从两个链的集合中分别选出一些不相交的链，最大化收益之和。 | 首先通过初步分析可知，由于选择链时的限制条件只存在于两个集合之间，因此结合数据范围容易想到二分图模型。题中的限制条件表现为某些特定的左点与右点不能同时被选择，收益即可视作相应的点权，那么题目所求即为最大化点权之和。不难发现，上述转化其实就是经典的二分图最大点权独立集模型，用网络流解决即可。 | 时O(M3)  空O(NlogN+M2) |
| Codechef  JUN 14 | Sereja and Arcs | 给定依次排列在一条直线上的N(N <= 100)个点，任意一对同色点之间都恰好连了一条圆弧，求有多少对异色圆弧相交。 | 利用补集转化，异色圆弧对总数减去形如AABB和ABBA的异色圆弧对数即为所求。用扫描线的方式从小到大枚举左端点x，然后再枚举右端点y，则方案数的计算可以借助后缀和求得，而加入一条圆弧的操作也能一并进行。维护后缀和即可。 | 时O(N)  空O(N) |
| Codechef  JAN 12 | Card Shuffle | 初始为1~N(N <= 105)的数列，M(M <= 105)次操作，每次取出一段数翻转并插入一个位置，求最终序列。 | 维护一棵平衡树，每次操作取出一个区间，打上翻转标记，再插入到指定位置中，最后dfs一遍输出。这样的问题通常采用Splay、Treap来维护，不过由于保证数据随机生成，我们可以直接把要访问的点旋到根。 | 时O(N+MlogN)  空O(N) |
| Codechef  JAN 12 | Misinterpretation 2 | 求长度在[L, R](R - L <= 50000, R <= 1010)，满足所有偶数位字符取出和所有奇数位字符相接与原串相等的小写字母串个数。 | 预处理部分，我们要筛出根号R内的质数，将p-1分解因数，并求出2模它们的幂的阶，并把所有质数分给它们的倍数。计算部分，我们要先将所有X+1分解质因数，分解过程中可能产生一个大质数，需要计算2模它的阶，之后再枚举X+1的所有因子。 | 时O(log2R)  空O() |
| Codechef  JUL 14 | Sereja and Equality | 求所有N(N <= 500)元排列的二元组中，包含数字相同且逆序对不超过E的段数之和。 | 可以使用动态规划求解。dp[x][E]表示长度为x的排列中，逆序对为E的排列数。dp[x][E] = 用前缀和优化即可。 | 时O(N3)  空O(N3) |
| Codechef  JUL 14 | Game of Numbers | 两个长度为N(N <= 400)的数组，每次从两个数组中同时取出不能完全重复的且不互质的两对数，满足大小关系不相同，且两对数的最大公约数不互质。问最多操作次数。 | 我们发现一对数对(i, j), (p, q)能否被加入，与当前的S1, S2无关，至与两个数组有关。建立一个二分图，每边N2个点。我们新建若干个点表示质数，把每一个数i向整除它的质数连边，容量为1；把每一个质数向它整除的数j连边，容量为1。将数组中相等的元素合并即可。 | 时O(N3)  空O(N3) |