试题泛做表格

by 赵亮

一、非challenge题

1.
试题编号：codechef Jan 2014
试题名称：CNTDSETS

题目大意：

求N维空间中，切比雪夫距离意义下，有多少个本质不同的点集直径等于D。
两个点集本质相同当且仅当它们可以通过互相平移得到。
N<=1000, D<=10^9, cases<=10

算法讨论：

直径恰好为D的难以计算，所以我们将其转化为直径不超过D的。
用直径不超过D的数目减去直径不超过D-1的数目即得到答案。
既然直径不超过D又能平移，则可以将每维坐标限制在[0,D]的范围内。
然而，这样并不能完全避免重复计数。
本题中，每个维度都可以集体加一个数，减一个数。
所以我们规定每个维度至少有一个0，就可以避免重复计数。
然而“至少有一个0”导致计数困难。
正难则反，不妨考虑使用容斥原理，转化成“某些维度一定没有0”，乘上容斥系数后求和即可。
即 sigma_{k=0}^n (-1)^k * C(N, k) * 2^(D^k * (D+1)^(N-k))
使用快速幂计算，预处理组合数。

时空复杂度：

时间复杂度：O(N^2+TNlogMOD)
空间复杂度：O(N^2)

2.
试题编号：codechef Jan 2014
试题名称：TAPAIR

题目大意：

给定一个N个点M条边的无向简单图，求有多少对边删去后整张图不连通。
N<=100000, M<=300000

算法讨论：

先处理边界情况：若原图不连通，则任一对边均符合要求。
不妨先求出图的一颗生成树。
求出生成树后，就分为两种情况（因为删去两条非树边显然仍然连通）：
(1)删去一条树边一条非树边
(2)删去两条树边
注意到一条非树边若连接树上两点u,v，则它可以替代u,v之间所有的树边。

为了讨论方便，我们不妨给每条非树边染上不同颜色，然后对于一条非树边，在它能替代的所有树边上染上它的颜色。
(1)删去一条树边一条非树边
    a.该树边上没有颜色
        即该树边无可取代，删去后自然不连通
    b.该树边上只有一种颜色，即该非树边的颜色
        删去了唯一能替代的非树边，自然不连通了
    （易知，若该树边上至少有两种颜色，则至少有两条非树边可以代替它，
    最多删去一条，图仍然连通）
(2)删去两条树边
    我们发现又有一条树边无色等多种情况
    为了方便，我们单独拎出“至少删了一条无色树边”的情况
    ###
        (0)至少删去一条无色树边
            显然会导致不连通
    ###
    下面我们考虑删去两条有色树边的情况
        a.两条树边的色彩集合不同
            很显然，可以分别取一条非树边“补”好，故不符合要求
        b.两条树边的色彩集合全等
            很显然，所有可以替代的非树边都跨过了这两条边，
            故删去后图一定不连通，符合要求
所以我们求出图的任意一颗生成树，染色，再按上述做法判断集合全等计数。
时间复杂度O(NM) 会超时，需要优化。
我们注意到我们只需要判断两个集合是否全等，这不禁让我们想到使用随机化算法。
给每种颜色一个随机数特征值，若干种颜色的并集的特征值即为它们的特征值的异或。
判断两个集合是否全等就只需判断它们的特征值是否相等，可以证明出错的概率非常小，可以忽略不计。
时间复杂度被优化到了O(N+M)，当然，使用哈希判断两个异或值的相等。
若使用平衡树或map判断，复杂度会上升到O((N+M)log(N+M))
要注意一条非树边连接树上两点u,v，设u,v的lca为w，则其需要给u-w,v-w均染上颜色。
有一个较简单的方法：使用dfs树作为生成树，这样的好处是：
因为无向图的dfs树只有树枝边和返祖边，所有的非树边都是返祖边，而一条返祖边只需染一段即可。
具体地，若一条返祖边从u返到v（v为祖先）
只需
```
w[u]^=Val;w[v]^=Val;
```
即可。

时空复杂度：

时间复杂度：O(N+M)
空间复杂度：O(N+M)

3.
试题编号：codechef May 2015
试题名称：CBAL

题目大意：

一个字符串平衡当且仅当它的每一个字符都出现了偶数次。Q次询问母串的某段子串中所有平衡子串长度的type次方和。
Len,Q<=10^5, type=0,1,2

算法讨论：

给每个字符分配一个不同的2的次幂的权值，一个子串的权值为其内所有字符的异或和，对异或和作部分和之后，L..R为平衡子串即等价于a_{L-1}=a_R。
然后分块维护，预处理两块间答案和块内所有对应权值（离散化后）坐标的type次方和，对于询问分成三段处理即可。

时空复杂度：

时间复杂度：O((Len+Q)*sqrt(Len))
空间复杂度：O((Len+Q)*sqrt(Len))

4.
试题编号：codechef May 2015
试题名称：GRAPHCNT

题目大意：

求有向图中存在多少点对(X,Y)满足存在一条1-X的路径与一条1-Y的路径，且这两条路径除点1外无交点。
N<=10^5, M<=5*10^5

算法讨论：

求出图以1为根的dominator tree，不合法点对即在根的同一棵子树的点对，总点对数（注意要以1能到达的总点数为基计算）减去不合法点对数即为合法点对数。

时空复杂度：

时间复杂度：O(Nalpha(N))/O(NlogN)
空间复杂度：O(N)

5.
试题编号：codechef Apr 2015
试题名称：BWGAME

题目大意：

求一个行列式的值，每行的1连续。
N<=10^5

算法讨论：

利用可并堆维护每个左端点对应的右端点集合，消元时对于一个左端点选择最左的对应的右端点，消去后将剩余部分与该右端点后1位的左端点合并即可。

时空复杂度：

时间复杂度：O(NlogN)
空间复杂度：O(N)

6.
试题编号：codechef Mar 2015
试题名称：RNG

题目大意：

求一个线性齐次递推数列的第n项，其中一项由前k项推得。
n<=10^18, k<=3*10^4

算法讨论：

初步想法是利用矩阵乘法优化，但复杂度至少达到O(k^3)，不可取。
故尝试使用化零多项式、特征多项式等优化方法，用FFT加速之后可以AC，详见郭晓旭的一篇相关论文。

时空复杂度：

时间复杂度：O(klog^2)
空间复杂度：O(k)

7.
试题编号：codechef Feb 2015
试题名称：DEVLOCK

题目大意：

求有多少个各位数字和不超过M且是P的倍数的N位数，对于M从0..MM都输出答案。
N<=10^9, P<=50, MM<=15000

算法讨论：

简单递推会超时，故考虑分治后用FFT合并，但问题在于DFT与inverse-DFT次数过多后常数巨大，无法通过；故尝试减少DFT与inverse-DFT次数，如两多项式相加时直接将点值相加（注意k次方不能这么做，理由显然，仅仅一部分点值无法代表全部），尽力常数优化可以AC。

时空复杂度：

时间复杂度：O(P^2MMlog)
空间复杂度：O(PMM)

8.
试题编号：codechef Nov 2014
试题名称：FNCS

题目大意：

在一个长度为n的数组上有标号为1..n的一个区间，一个区间的权值为区间内所有数权值的和。维护m次操作，每次修改一个数或者询问编号连续的一段［区间］的［区间权值］［的］［和］。
n<=10^5, m<=10^5

算法讨论：

有经典数据结构题方法“每若干次暴力重建可持久化数据结构”的方法，这里可以沿用。
我们每若干次就暴力重新计算所有区间的权值，并重新维护。
我们再对数组分块，修改整块时暴力更新。
那么只剩下询问零散的情况，因为只有维护前缀和才能快速询问，故对下标分块。
由上知所有块的大小均取sqrt最合适，时空复杂度最低。

时空复杂度：

时间复杂度：O(Nsqrt(N))
空间复杂度：O(Nsqrt(N))

9.
试题编号：codechef Sept 2014
试题名称：QRECT

题目大意：

在平面中维护m次操作，每次插入一个矩形，删除一个矩形，或者询问与某个特定矩形有交集的矩形个数。
m<=10^5

算法讨论：

可以发现删除矩形是很难实现的，这会破坏已有数据结构的维护能力。
所以尝试使用cdq分治，将“删除矩形”转化为“矩形生命周期”。
这样就只剩下询问，乍一看需要很复杂的容斥，然而我们发现“有交集”这一看似麻烦的条件反而只需要用右端点减去左端点即可。
也可在“时间线段树”上dfs。

时空复杂度：

时间复杂度：O(mlog^2)
空间复杂度：O(m)

10.
试题编号：codechef Apr 2014
试题名称：GERALD08

题目大意：

一棵n个节点的树中，边分红、蓝两色。游戏者A每轮删一条红边，游戏者B每轮删一条蓝边。（删除任意一条边后，与根不连通的部分将被移除）谁先不能操作谁输。问A先手时谁胜，以及B先手时谁胜。
n<=10^5

算法讨论：

需要用到高深的博弈理论。
这里脱离“SG函数”的想法，因为SG函数针对的是公平博弈，而此处明显是不公平博弈（不对等博弈）。
所以我们用一个状态的两个“最优”后继（A走后局面与B走后局面）来夹逼这个状态，得到一个小数描述，且分母一定为2的次幂（证明详见官方题解，略）。
然而暴力计算很显然会超时，所以考虑利用树的性质，对于一棵子树，计算每个孩子对它的权值的贡献，而每个孩子的贡献按照所连边的颜色分类讨论即可。
因为是高精度小数（最多可达到n位小数）故可以考虑用平衡树维护每一位、启发式合并做运算。
最后若结果为正则意味着“夹逼”的两个都非负，即A无论如何必胜，为负则恰恰相反，B必胜。
结果权值为0意味着“夹逼”的两个一正一负，故后手胜。

时空复杂度：

时间复杂度：O(Nlog^2N)
空间复杂度：O(NlogN)

11.
试题编号：codechef July 2015
试题名称：EASYEX

题目大意：

有一个K面的完全公平骰子，投N次，求前L个面投到次数的F次方和的期望。
N,K<=10^9, 1<=L<=K, F<=10^3, L*F<=5*10^5

算法讨论：

由于期望的线性性，即期望的和等于和的期望，故可以将每项展开分别求系数。
同样用递推法求系数。
上面我写的第7题中的“FFT加速”在原理上还是可以的，但不能这么做。
考虑到分治并无意义，所以我们直接对这个递推式转化成多项式。
然后使用FFT加速计算。
算出系数后累加答案即可。

时空复杂度：

时间复杂度：O(LFlog^2)
空间复杂度：O(LF)

12.
试题编号：codechef Aug 2013
试题名称：PRIMEDST

题目大意：

在一棵n个节点的树中随机选一条路径，求路径长度为质数的概率。
n<=5*10^4

算法讨论：

这是一道经典题。
不过这是作为一道树的点分治的经典题。
然而我们发现，“路径长度为质数”根本不是什么特殊性质，只能暴力统计所有路径长度的路径数量，最后利用古典概型计算。
在点分治暴力统计中涉及到卷积，暴力计算会TLE，故使用FFT加速卷积的计算。

时空复杂度：

时间复杂度：O(Nlog^2N)
空间复杂度：O(N)

13.
试题编号：codechef Mar 2015
试题名称：TREECNT2

题目大意：

n个节点的树维护m次操作，每次修改一条边权。定义一条路径的权值为路径上所有边权值的最大公约数。每次操作后输出树中路径权值为1的路径条数。
n<=10^5, m<=100, 边权不超过10^6

算法讨论：

权值恰为1的路径条数十分难求，所以我们考虑莫比乌斯反演。
我们先求出对于每个d，权值为d的倍数的路径条数，然后再将其分别乘上反演系数，即可求出权值恰为1的路径条数。
由于路径权值的定义，只需用并查集维护若干连通关系即可。
然而考虑到还要“断”，所以用“启发式合并”的方法实现一种类似于“可持久化并查集”的结构，上面的连接带有时间戳。

时空复杂度：

时间复杂度：O(边权最大值*(N+M^2)log)
空间复杂度：O(边权最大值*N)

14.
试题编号：codechef Jan 2015
试题名称：RANKA

题目大意：

在9*9的围棋棋盘上构造n步棋。
n<=10^4

算法讨论：

构造8个连环劫，每次遍历一遍连环劫即可下256手（类似grey码）。
每次遍历之后找一个劫材，可以迅速走满n步棋。

时空复杂度：

时间复杂度：O(N)
空间复杂度：O(1)

15.
试题编号：codechef Jan 2015
试题名称：XRQRS

题目大意：

一个空序列维护m次操作；操作种类：（1）末尾加一个数（2）末尾删k个数（3）询问区间k小值（4）询问区间与x异或后结果最大的数（5）询问一个特定区间中数x的排名。
m<=5*10^5

算法讨论：

利用compacted_trie
直接用可持久化trie维护整个序列（思想类似函数式线段树求区间k小值）
（3）（5）互为逆操作，只需在trie上走、前缀求和
（4）是trie的经典做法
故所有操作都能维护

时空复杂度：

时间复杂度：O(mlogm)
空间复杂度：O(mlogm)

16.
试题编号：codechef Dec 2014
试题名称：RIN

题目大意：

n门课m个学期，课程之间有若干“先修”关系。第i个学期选修j可以得到X[i][j]的分数。求学完这n门课可以获得的最多分数。
n,m<=100

算法讨论：

拆点最小割模型。
每门课拆成m个点，依次相连，按照“其中至少割一条”建边。（即链式建边）
注意到目标函数是总分数最大，所以我们取反再加一个足够大的值即可。
对于先修关系，假设a是b的先修课，则可以由a的第i学期向b的第i+1学期连边权为正无穷的边，即若a在第i学期后学习，b也必须在第i+1学期后学习。对于每个i都这么建边，就可以保证“先修”关系。

时空复杂度：

时间复杂度：O(maxflow(N,M)) N=nm M=nm
空间复杂度：O(M) M=nm

17.
试题编号：codechef Oct 2014
试题名称：TRIPS

题目大意：

一棵n个节点的树每条边权值为1或2。Q组询问一个点走到另一个点要几天（每天最多走一个定值且必须停在节点上）。
n,Q<=10^5

算法讨论：

对于定值大于sqrt(n)的询问，步数必小于sqrt(n)，每步倍增暴力即可，复杂度O(Qsqrt(n)log)
对于定值小于sqrt(n)的询问，我们直接预处理每个点该定值下一步在哪里，这个“终点”我们记为每个点的“虚父亲”，在“虚父亲”构成的树上倍增模拟即可，复杂度仍为O(Qsqrt(n)log)

时空复杂度：

时间复杂度：O(Qsqrt(n)log)
空间复杂度：O(nlog)

18.

试题编号：codechef Aug 2013
试题名称：LYRC

题目大意：

给定n个模式串和m个母串，对于每个模式串，求它在所有母串中的总出现次数。
n<=500， 对应串长<=5000， m<=100， 对应串长<=50000

算法讨论：

对所有模式串建AC自动机，将所有母串在其上运行一遍，然后对每个模式串对应节点的fail子树分别统计求和即可。
注意：本题若用后缀数组做，会TLE。

时空复杂度：

时间复杂度：O(n*对应串长+m*对应串长)
空间复杂度：O(n*对应串长)

19.
试题编号：codechef Aug 2014
试题名称：SIGFIB

题目大意：

输入n,m；要求在模m意义下求
sigma_{6xyz*fib[x]*fib[y]*fib[z]*[x+y+z=n]}
其中fib[x]为斐波那契数列第x项
n<=10^18, m<=10^5

算法讨论：

利用斐波那契数列关于模m的周期加速计算（实践证明周期很短）即可。

时空复杂度：

时间复杂度：O(m)
空间复杂度：O(m)

20.
试题编号：codechef July 2014
试题名称：GNUM

题目大意：

输入两个长为N的数组A,B，每轮要选出两个数对(i,j),(p,q)满足
（1）(i,j)不在集合S1中，(p,q)不在集合S2中
（2）A_i<B_j, A_p>B_q
（3）gcd(A_i,B_j,A_p,B_q)>1
选完后(i,j)加入集合S1，(p,q)加入集合S2
求最多可以进行多少轮
n<=400

算法讨论：

选出所有gcd(A_i,B_j)>1的数对(i,j)
分成左边（A_i<B_j）和右边（A_i>B_j）两部分
左右两点之间再做一次gcd，若大于1则连边
然后求此二分图的最大匹配即为答案
但是这样会TLE
所以要想办法根据gcd的特殊性质优化。
注意到这个最大匹配等价于网络流的最大流（点容量、边容量均设为1时）。
我们可以对于每个质数p新建一个点容量无穷的“虚点”
左边gcd为p的倍数的点连向虚点p
右边gcd为p的倍数的点从虚点p连入
注意到这样和原图是等价的，却大大减少了边数
这样就可以在时间限制内AC了

时空复杂度：

时间复杂度：O(maxflow(n^2,n^2log))
空间复杂度：O(n^2log)

21.
试题编号：codechef July 2014
试题名称：SEAEQ

题目大意：

两个长度为N的数组A,B被称作相似的，当且仅当每个对应下标所对应的值在对应数组中的排名相等。关于两个长度为N的数组的一个函数f值为某个下标区间的个数，该下标区间需要满足在两个数组中分别对应该下标区间的那一段相似且逆序对数不超过一个定值E。
回答t组询问，每个询问要求你求一个F值的总和，并且两个数组均为1-n的排列。
t<=10^4, N<=500, E<=10^6

算法讨论：
先预处理长度为i的排列逆序对数不超过E的方案数（通过递推）。
然后枚举给F函数作出贡献的那段区间的长度，显然长度为L的区间有(N-L+1)个可能位置，对区间内和区间外分别统计数量，乘以贡献求和统计答案即可。

时空复杂度：

时间复杂度：O(n^3)
空间复杂度：O(n^3)

22.
试题编号：codechef June 2014
试题名称：TWOCOMP

题目大意：

今有一棵n个节点的树与两个带权路径集合A,B，要求在两个集合中分别选出一个子集使得原属于不同集合的被选出的路径不相交，并使得选出的所有路径权值总和最大。
n<=10^5, |A|,|B|<=700

算法讨论：

很显然是二分图最大点权独立集，转化为二分图最大权匹配。
唯一需要处理的是建边的问题，把一条路径拆分成上下两条，就可以根据dfs序求交。

时空复杂度：

时间复杂度：O(maxflow(|A|+|B|,|A||B|))
空间复杂度：O(|A||B|)

23.
试题编号：codechef Feb 2013
试题名称：QUERY

题目大意：

给定一棵n个节点的树，要求维护以下操作：
（1）路径加等差数列
（2）路径求和
（3）恢复历史版本
n,m<=10^5

算法讨论：

很显然先树链剖分，然后线段树维护即可。
等差数列可以直接转化为标记，并且可合并。
支持恢复历史版本，只需将线段树改为可持久化线段树即可。

时空复杂度：

时间复杂度：O(mlog^2n)
空间复杂度：O(mlog^2n)

24.

试题编号：codechef Nov 2012
试题名称：COUNTARI

题目大意：

输入一个长度为N的数组A，统计有多少对(i,j,k)满足A_i+A_k=2A_j
(i<j<k)
n<=10^5 A_i<=3*10^4

算法讨论：

这和一道FFT经典题十分相似，但仔细想却无法用FFT直接解决。
故我们考虑先分块。
分块后，
（1）至少存在一块包括i,j,k中的至少2个，则直接以块为单位统计即可
（2）i,j,k均在不同块中，枚举j所在块即可FFT卷积

时空复杂度：

时间复杂度：O(nsqrt(nlog))
空间复杂度：O(nsqrt(n))

25.
试题编号：codechef Dec 2013
试题名称：QTREE6

题目大意：

一棵n个节点且每个节点为黑或白色的树维护m次操作
每次操作
（1）给一个节点改色
（2）假设树上只剩连接同色点的边，询问一个节点所在的连通分量的大小
n,m<=10^5

算法讨论：

对于黑白二色分别维护一个树链剖分。
改色时在两棵树上更新路径，复杂度可以均摊。
询问就可以直接用“一个节点记他虚儿子的信息”合并了。

时空复杂度：

时间复杂度：O(nlog^2)
空间复杂度：O(n)

26.
试题编号：codechef May 2014

试题名称：ANUDTQ

题目大意：

一棵n个节点的带点权的树（1为根）维护m次操作
每次操作
（1）添加一个给定权的叶子
（2）删除一棵子树
（3）子树点权加
（4）子树点权求和
强制在线
$n,m<=10^5$

算法讨论：

由于所有操作几乎都是修改子树、询问子树，
而子树修改在涉及到树形态变化时比较复杂。
所以我们转化一步，相当于序列问题中的“前缀和”或“差分”
每个节点记它子树的信息
这样问题就转化为了修改一条链、打标记、询问一条链或一个点
而这些都可以用LCT解决

时空复杂度：

时间复杂度：O(nlogn)
空间复杂度：O(n)

27.
试题编号：codechef June 2014
试题名称：SEAARC

题目大意：

给定一个长度为n的数组A，对于每一对<i,j>(i<j)若满足$A_i=A_j$，就在坐标系中画一个以(i,0)~(j,0)为直径的圆，问有多少对颜色不同的圆存在交点。
$n<=10^5$, $A_i<=10^5$

算法讨论：

直接计数有一点困难，我们考虑正难则反。
用总对数减去外离的圆和内含的圆即可得到相交的圆对数。
外离的圆很好统计，以下标为单位分别统计即可（一边“紧”一边“松”）。
内含的圆我们使用类似“分块”的做法，分成出现次数大于某个阈值的颜色和不大于某个阈值S的颜色分别统计即可。

时空复杂度：

时间复杂度：O(nS)
空间复杂度：O(n)

28.
试题编号：codechef May 2014
试题名称：SEINC

题目大意：

输入两个模4意义下的数组A,B，每次可以给A的一段区间+1，问多少次能把A变成B（数组长度为n）
n<=10^5

算法讨论：

先对A,B作差再差分，问题就转化为可以给l位+1，r位-1，问最少多少步。
由于是模4意义，所以可以对2和3贪心。

时空复杂度：

时间复杂度：O(n)
空间复杂度：O(n)

29.
试题编号：codechef Mar 2014
试题名称：STREETTA

题目大意：

给定两个长度为n的数组A,B，每次给A区间加一个等差数列或者给B区间等差数列取max，并且会询问A,B中的点值，维护以上操作m次。
n<=10^9, m<=3*10^5

算法讨论：

和线段树打标记的经典题十分类似。
区间加等差数列很显然，只需要记录首项和公差。
区间等差数列取max的话两个数列必然有一个“拐点”。
不失一般性，设拐点在左子树。
则把一般数列放在当前节点，拐点左侧的数列下传到 左子树即可。
这样时间复杂度要额外再乘以一个log（本来应该被“修改”到的每个点会对应一条链下去）
另外，n很大，所以此题需要动态开节点，所以空间复杂度也要乘以log

时空复杂度：

时间复杂度：O(mlog^2n)
空间复杂度：O(mlogn)

30.
试题编号：codechef Mar 2014
试题名称：GERALD07

题目大意：

输入一个n个点m条边的无向图，Q组询问如果只保留编号在[L_i,R_i]范围内的边，则图中有多少个连通分量。
n,m<=2*10^5, q<=2*10^5

算法讨论：

从小到大枚举询问的右端点，以边的编号为边权，很显然要尽量先用编号较大的边
所以我们维护最大生成森林，森林中边权>=L_i的数量作为n的减数，即得到连通分量的个数。

时空复杂度：

时间复杂度：O(nlogn)
空间复杂度：O(n)

31.
试题编号：codechef Feb 2014
试题名称：DAGCH

题目大意：

输入一个n个点m条边的无向图，每一个点标号为其dfs序，且1号点可以到达所有点。
点x对于点y是最好的当且仅当x是满足存在一条p到y的路径且路径上每个中间节点标号都大于y的点p中标号最小的那一个。
Q组询问对于每个点，它对于多少个点是“最好的”。
n<=10^5, m<=2*10^5, Q<=10^5

算法讨论：

很显然“最好”的点即为dominator tree的半必经点。
运行Languer-Tarjan算法即可。

时空复杂度：

时间复杂度：O(nalpha(n))
空间复杂度：O(n)

32.
试题编号：codechef Feb 2014
试题名称：COT5

题目大意:

一棵初始为空的treap，维护n次操作:
（1）插入一个给定权值和重量的节点
（2）删除一个节点
（3）询问两节点之间的路径长度
（保证权值、重量都互异，即treap的形态唯一）
n<=2*10^5

算法讨论:

和rmq转lca问题类似，两个节点的lca即为他们的dfs序间隔中重量最大的节点。
求得lca后只需要即时地求一个节点的深度，而这可以通过线段树维护。

时空复杂度;

时间复杂度：O(nlog^2n)
空间复杂度：O(n)

33.
试题编号：codechef Dec 2013
试题名称：REALSET

题目大意:

输入一个长度为n的整数数组A，问是否存在一个不全为0的长度同样为n的数组B满足B任意轮换之后与A的点积均为0。
n<=3*10^4

算法讨论:

问题可以转化为判断一个矩阵是否非奇异，简单判断即可。

时空复杂度:

时间复杂度：O(nlogn)
空间复杂度：O(n)

34.
试题编号：codechef Nov 2013
试题名称：MONOPLOY

题目大意:

输入一棵n个节点的有根树，点有点权。
维护m次操作:

（1）把点到根的路径上的点权都改成同一个值
（2）询问一个点所在子树中所有点到根的路径上的权值总数的平均值
n,m<=2*10^5

算法讨论：

我们记录每个值向上重复延伸的最上方一个点，类似于LCT的expose操作的势能分析保证复杂度。
平均值显然只需要dfs序+线段树维护即可（因为是子树和）。

时空复杂度：

时间复杂度：O(nlog^2n)
空间复杂度：O(n)

35.
试题编号：codechef Oct 2013
试题名称：FN

题目大意：

输入M,C，输出在模M意义下斐波那契数列中第一个值为C的项的下标。
模数M为质数且尾数为0,1,4,9之一。
M<=2*10^9

算法讨论：

在题目条件下2存在逆元，5也存在二次剩余，即sqrt(5)存在且可用。
此时只要利用斐波那契数列通项公式列方程，解方程式将“开根”转化为求二次剩余。
（求二次剩余用离散对数，即大步小步法）
解出下标即可。

时空复杂度：

时间复杂度：O(sqrt(M))
空间复杂度：O(sqrt(M))

36.
试题编号：codechef Sept 2013
试题名称：TWOROADS

题目大意：

二维平面上有n个点，要构造两条直线使得所有点到两条直线的较近距离的平方和最小。
求构造两条直线。
（不存在重点、三点共线）
n<=100

算法讨论：

假设知道两条直线分别的控制范围，则可以轻松地通过代数计算算出。
易知控制范围由角平分线分割。
大胆猜测角平分线必过两点，枚举即可。
证明可以用微量偏移法。

时空复杂度：

时间复杂度：O(n^4)
空间复杂度：O(n)

37.
试题编号：codechef Sept 2013
试题名称：TMP01

题目大意：

一个空串维护m次操作：
（1）push_back
（2）pop_front
（3）询问当前串内不同子串个数
n<=10^6

算法讨论：

很显然是后缀自动机的经典问题。
然而本题使用后缀自动机会MLE，故不妨用map存储转移边压缩空间，以时间换空间。

时空复杂度：

时间复杂度：O(nlogALPHA)
空间复杂度：O(nlogALPHA)

38.
试题编号：codechef July 2013
试题名称：RIVPILE

题目大意：

输入n个点m种价格不同的圆。为每个点购买至多一个圆使得至少存在一条路径从直线y=d到直线y=0，路径中只经过圆上的点（即只走圆的边界）。
n,m<=250

算法讨论：

每个点拆m个点之后，根据圆边界的相交情况运行最短路算法即可。
用类似“前缀和”的方法优化边数。
采用堆优化的dijkstra算法实现最短路。

时空复杂度：

时间复杂度：O(n^2mlogn)
空间复杂度：O(n^2m)

39.
试题编号：codechef June 2013
试题名称：TKCONVEX

题目大意：

提供n根长度已知的木棍，求能否拼出2个边数为k的凸多边形（每根木棍总计最多用一次，可以不用）。
n<=1000, k<=10

算法讨论：

先考虑较简单的情况，假设只拼1个凸多边形。
注意到凸多边形的唯一条件是最长边长度小于其它边长度之和，那么肯定是选取连续的一段。
所以推广到拼2个凸多边形的情况，可以发现要么取连续的2段，要么取连续的一大段（长度为2k），分配给两个凸多边形。
分情况讨论即可。

时空复杂度：

时间复杂度：O(C(2k,k))
空间复杂度：O(n)

40.
试题编号：codechef June 2013
试题名称：SPMATRIX

题目大意：

一个n*n的方整数矩阵是特别的当且仅当
（1）主对角线全为0
（2）其余所有元素均在1..(n-2)之间
（3）对于所有0..(n-2)，矩阵中均存在一个元素与其等值
（4）A_{ij}<=max(A_{ik},A_{kj})
求n*n的特别矩阵个数
n<=10^7

算法讨论：

直接用组合数学方法推出公式即可，而后直接计算。

时空复杂度：

时间复杂度：O(n)
空间复杂度：O(n)

41.
试题编号：codechef May 2013
试题名称：QTREE

题目大意：

输入一棵n个点的环+外向树（环长为奇数）。
边有权，但两点间最短路定义为两点间经过的边的［数量］最少的路径。
维护m次操作：
（1）两点间最短路径上的所有边权取相反数
（2）两点间最短路径上所有边权询问最大子段和
n,m<=10^5

算法讨论：

对外向树维护树链剖分，对基环维护一棵线段树。
很显然相反数是可以打标记的同时维护最大子段和的。

时空复杂度：

时间复杂度：O(nlog^2n)
空间复杂度：O(n)

42.
试题编号：codechef Mar 2013
试题名称：LECOINS

题目大意：

输入n种给定面值和类型的硬币，每种硬币均有无限个。
回答Q组询问：选出一些硬币的总面值为S时，最少用到多少不同类型的硬币。
n<=30, Q<=2*10^5, S<=10^18, 面值<=2*10^5

算法讨论：

以面值最小的硬币为基准硬币，设其面值为k
则一下所有计算均可以在模k意义下进行
那么其后直接dp即可

时空复杂度：

时间复杂度：O(n^2*最大面值范围)
空间复杂度：O(n*最大面值范围)

43.
试题编号：codechef Jan 2013
试题名称：ANDOOR

题目大意：

输入一个矩形范围和n个圆，求n个圆的并的周长在矩形范围内的总长度。
n<=1000

算法讨论：

直接对于每个圆求出其在矩形内部且未被其它圆覆盖的周长长度
将每个圆的上述“长度”累加求和即可。

时空复杂度：

时间复杂度：O(n^2log)
空间复杂度：O(n)

44.
试题编号：codechef June 2015
试题名称：CHEFBOOK

题目大意：

输入一个查分约束系统，并最小化所有约束条件中代数式的和。
变量<=100, 约束<=10000

算法讨论：

查分约束系统很显然用最短路算法解决。
但此题需要最小化代数式的和，很显然直接用最小费用最大流算法即可。
一个理解提示：最小费用最大流算法本来就是不停运行最短路算法找增广路。

时空复杂度：

时间复杂度：O(maxflow(变量,约束))
空间复杂度：O(max(变量,约束))

45.
试题编号：codechef Jan 2013
试题名称：CUCUMBER

题目大意：

输入B个n*n的矩阵A_i，
对于数对(a,b)定义n*n的矩阵B满足
B_{ij}=sigma_{k=1}^{n} A_{aik}*A_{bjk}
一个1-n的排列P是特别的当且仅当存在一个i满足
B_{iP_i}是奇数
数对(a,b)是特别的当且仅当特别的排列有奇数个
求有多少个数对是特别的
n<=60, B<=8000

算法讨论：

令C_{ij}=(B_{ij}+1) mod 2
排列P是特别的当且仅当它对方阵C的行列式值贡献为0
非特别排列的贡献均为1或-1
故det(C)的奇偶性即为非特别排列数的奇偶性
当n>1的一般情况，总排列数为偶数
故det(C)的奇偶性即为特别排列数的奇偶性
故数对(a,b)是特别的当且仅当det(C)为奇数
在每个A_i后补上全为1的1行，B即为一个A_i与另一个A_j的转置相乘
利用代数余子式等方法消元
但是这样会超时，所以采用压位优化

时空复杂度：

时间复杂度：O(n^2B+B^2)
空间复杂度：O(n^2B)

46.
试题编号：codechef Dec 2012
试题名称：DIFTRIP

题目大意：

输入一棵n个节点的树，每个节点的权值定义为其度数，两条路径全等当且仅当将经过的点的权值依次连成一个数列后两个数列全等。
求有多少条从儿子到祖先的不同路径。
n<=10^5

算法讨论：

直接使用后缀自动机。
与后缀自动机求不同串数完全类似，只需改成树上即可。

时空复杂度：

时间复杂度：O(nlogALPHA)
空间复杂度：O(nlogALPHA)

47.
试题编号：codechef Nov 2014
试题名称：SEAORD

题目大意：

n个程序2台电脑，第i个程序在2台电脑上分别要运行A_i, B_i秒。
现在要在最短时间内运行完所有程序（2台电脑上各运行一遍），求最短时间+方案。
限制条件是2台电脑不能同时运行同一程序。
n<=10^5

算法讨论：

题解中的一个结论：
很显然max(sigmaA_i, sigmaB_i, A_i+B_i)是答案的下界
但可以证明下界一定可以取到。
发现最优方案非常多，故可以多次随机直到随机出最优方案为止。

时空复杂度：

时间复杂度：无法估计
空间复杂度：O(n)

48.
试题编号：codechef Oct 2011
试题名称：BAKE

题目大意：

维护n个操作，每个操作都是增加订单或询问：
每个订单：
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
年龄不超过90，出售数不超过100，括号内内容可能被省略
询问同理，询问缺省部分表示该部分无限制，询问一个年龄区间的总出售数
各维大小均为常数
n<=100000

算法讨论：

直接模拟
记前缀和即可

时空复杂度：

时间复杂度：O(n)
空间复杂度：O(n)

49.
试题编号：codechef Oct 2011
试题名称：PARSIN

题目大意：

输入n,m,K
求sigma_{k=1}^{n} sigma_{i=1}^{m} sin(k_iK)
n<=10^9, m<=50

算法讨论：

简单二维递推即可求出答案
但是很显然会超时
所以矩阵乘法优化即可

时空复杂度：

时间复杂度：O(m^3log)
空间复杂度：O(m^3)

50.
试题编号：codechef Aug 2015
试题名称：DISTNUM

题目大意：

长度为n的数列维护m次操作：
（1）令区间[l,r]中出现过的数字的集合为S，询问S中所有三元组乘积的和模一个大质数。
（2）插入一个数
（3）删除一个数
（4）修改一个数
（5）询问一个区间内出现过多少个不同数字
n,m<=10^5

算法讨论：

先用平衡树离线预处理一遍给每个数分配一个不同的下标，
然后从头开始做。
（即解决了插入、删除的问题）
然后参考"spoj XXXXXXXX"的做法，每个数记(上次出现的位置，值)
然后就可以树套树维护了

时空复杂度：

时间复杂度：O(mlog^2)
空间复杂度：O(mlog^2)

51.
试题编号：codechef Sept 2012
试题名称：KNGHTMOV

题目大意：

一个无穷大的棋盘上，骑士的起点为(0,0)，骑士可以从(x,y)移动到(x+A_x,y+A_y)或(x+B_x,y+B_y)。
有k个障碍物，骑士不能移动到障碍物上，问有多少种方案把骑士移动到(X,Y)。
答案模一个大质数。
k<=15, |其他数字|<=500

算法讨论：

若两个向量不线性相关，则每个位置都可以用这两个向量线性表出，求组合数即可。
障碍物O(2^k)容斥即可。
若两个向量线性相关，则可以把能到达的所有位置映射到一维，之后递推即可。

时空复杂度：

时间复杂度：O(k^2*2^k)
空间复杂度：O(k^2)

52.
试题编号：codechef Aug 2012
试题名称：MAGIC

题目大意：

输入n个点m条边的无向简单图，两个人的起点分别为1号点和2号点。
从第一个人起轮流操作，每次操作有3步：
（1）可以移动到同一连通块的任意节点，如果与对方相撞则己方胜利
（2）加一条无向边，若无法加入则对方胜利
（3）最开始每人p次传送机会，若传送机会有剩余则可以消耗一次传送机会将自己传送到任意一个节点
问最优策略下胜方
n<=7777, m<=10^5, p<=10^5

算法讨论：

注意到每人最多传送一次，然后就直接递推了。

时空复杂度：

时间复杂度：O(n+m)
空间复杂度：O(n+m)

53.
试题编号：codechef Aug 2012
试题名称：GTHRONES

题目大意：

有n个不同的数字，第i个是u_i，出现了c_i次。
二人轮流操作。
第一个人选择一个数字作为初试的特征值。
之后每轮需要选出一个与原特征值相似的新特征值，无法操作时算输。
原特征值被移出（若有多个仅移出一个）。
两个数被认为是相似的当且仅当一个是另一个的倍数且商是质数。
求最优策略下胜方。
若先操作者胜，还需要输出先操作者在胜利的前提下初始特征值最大能选几。
n<=500, u_i<=10^18, c_i<=10^9

算法讨论：

对于相似的两个数连一条边
（判断质数用miller-rabin算法）
由“相似”的性质，知图中必然没有环
当然更没有奇环，所以一定是二分图
而这就转化到了一道经典的博弈题（即在棋盘上移动棋子，谁先不能移动为输家）
这不禁让我们想到同样套用该经典题的算法
若二分图有完备匹配，则后手胜，否则先手胜
而当先手胜时，即为一个最大匹配是否“必须”包含某条边
退流或变成有向边后做均可

时空复杂度：

时间复杂度：O(maxflow(n,nlogn))
空间复杂度：O(nlogn)

54.
试题编号：codechef July 2012
试题名称：DGCD

题目大意：

输入n个节点的点上带权的树，
维护m次操作：
（1）询问两点间路径上所有点权的gcd
（2）两点间路径上所有点权加一个值
n,m<=5*10^4

算法讨论：

注意到gcd(x,y,z)=gcd(x,y-x,z-y)
故只需要维护查分后的序列和原点权即可
通过树链剖分即可将树上问题转化为序列上问题。

时空复杂度：

时间复杂度：O(mlog^2nlog权值)
空间复杂度：O(n)

55.
试题编号：codechef May 2012
试题名称：TICKETS

题目大意：

n道菜m个人，晚宴时每道菜恰好分给一个人，第i个人会开心当且仅当吃到某两种菜。
求一个最大的k，使得任意k人参与晚宴，都有分配方案使得每个人都开心。
n<=200, m<=500

算法讨论：

每道菜建点，一个人喜欢的2道菜之间连边。
考虑只要存在一种情况使得某p个人无论如何也无法开心，答案就是p-1。
某p个人无论如何也无法开心即一个点导出子图边数至少比点数多1。
对这个点导出子图的形态分情况讨论即可。
一种是两点间三条路径，直接bfs。
一种是两环间一条路径，根据lca更新答案即可。

时空复杂度：

时间复杂度：O(n^2m)
空间复杂度：O(n+m)

56.
试题编号：codechef Jan 2012
试题名称：CARDSHUF

题目大意：

n张卡片一开始标号从1-n按标号从小到大摆放。
维护m次操作：
取出前A张，取出前B张，放回A张，取出前C张，逆序放回B张，放回C张
求最后顺序
n<=10^5, m<=10^5

算法讨论：

直接使用split-merge treap模拟即可
只需维护一个翻转标记

时空复杂度：

时间复杂度：O((n+m)logn)
空间复杂度：O(n)

57.
试题编号：codechef Apr 2012
试题名称：TSUBSTR

题目大意：

输入一棵n个节点的点上带一个小写字母的树。
一个字符串能被这棵树表示当且仅当存在某个树上节点往其后代走出的一条路径拼出后与该字符串全等。
第一问求有多少个字符串能被这棵树表示。
第二问给出m组询问，每组询问先给出一个26个字母的大小关系，
然后询问在此大小关系意义下所有能被这棵树表示的字符串中第k_i小的字符串。
n<=2.5*10^5, m<=5*10^4

算法讨论：

很显然对树建出SAM。
第一问就是求路径数。
第二问只要意识到路径和能被表示的字符串一一对应，就可以逐位确定了。
26个字母的大小关系在后缀自动机中并无意义。
这也是采用后缀自动机而非后缀数组的原因。

时空复杂度：

时间复杂度：O(mlogn*ALPHA)
空间复杂度：O(n*ALPHA)

58.
试题编号：codechef Apr 2012
试题名称：CONNECT

题目大意：

n*m的网格，每个格子拥有一个[-1,n*m]的整数特征值和一个花费。
求一个总花费最小的4连通块，块上所有格子特征值不为-1且至少有k种不同特征值。
n,m<=15, k<=7

算法讨论:

一个经典思想：将特征值向1-k进行随机映射。
然后只需状压dp即可。
卡时随机即可得到答案。

时空复杂度:

时间复杂度：无法估计
空间复杂度：O(nm)

59.
试题编号：codechef Dec 2011
试题名称：HYPER

题目大意:

一个超图类比普通的图，区别在于每条“边”都连接3个点。
一个超树是一个去掉任意一条边都不连通的连通超图。
带标号n个节点的超树计数。
n<=17

算法讨论:

在超树中搜索点双，砍掉叶子之后可以转化为普通图。
暴力枚举每个点双的大小，搜出拼出的超树数量。
由于会超时，故打表即可。

时空复杂度:

时间复杂度：O(n)
空间复杂度：O(n)

60.
试题编号：codechef Feb 2012
试题名称：FINDSEQ

题目大意:

输入一个长度为N的数组A和一个5的排列p，
求一个A的长为5的子序列B使得其满足:
序列B中恰好有(p_i-1)个数比B_i小
（即序列B的排名关系等于排列p）
n<=2000

算法讨论：

先枚举第二个数与第四个数的下标。
然后易知根据大小关系可以用upper_bound/max贪心得到第一个数与第五个数。
最后检查是否有符合要求的第三个数在对应范围内。

时空复杂度：

时间复杂度：O(n^2)
空间复杂度：O(n^2)

二、challenge题

1.
试题编号：codechef Jan 2014
试题名称：TOURBUS

题目大意：

给定一个N个点的无向平面图，要求用最少的合法路径覆盖它的每一条边恰好一次。
一条合法路径或者是一条简单路径，或者是一个环，并且不能自交（平面图意义）。

算法讨论：

这是一道challenge题，很显然此题是一个NP-Hard问题，所以我们着眼于找到一个近似方法/部分方法。
针对这种情况常用的方法有搜索、贪心、DP（错误DP）等等或它们的综合，无论使用哪种方法，都建议自己写一个check验证答案/统计最优性。
我们可以发现，一个朴素的解是尽量连上相邻的一对边，在平面图上相邻的两条边显然不会相交。
可以得知这样需要(N+M)/2条路径。
然而我们仍然对解题并没有太多头绪，所以尝试考虑缩小范围。
比如如果边数较小，我们可以采用状压DP的方法获得答案。
第一步，生成所有合法路径，标记为这些路径所对应bit的或。
第二步，对于任意边集S，通过子集枚举的方法dp出覆盖S上的每条边恰好一次所需的最少路径数。
子集枚举的时间复杂度是O(3^M)的，可以处理M<=15的情况。
伪代码：
```
f[0]=0;
for (int S=1;S<2^M;S++){
	f[S]=+inf;
	for (sub=S;sub>0;sub=S&(sub-1)){
		if (S/sub is valid){
			f[S]=min(f[sub]+1,f[S]);
		}
	}
}
```
对于较大的范围，我们不妨先采取贪心的策略。
例如，不断找到最长的路径并删除。
但是我们发现，找到最长路径仍然是一个NP-Hard问题。
我们尝试使用一个启发式方法：不断加长当前路径。
我们尝试加入一条边之后要与已有的边判交（平面图意义），如果不能通过，也可以考察加入这条边之后哪些边不可行了，根据拓扑关系顺列边。

我们还可以随机断掉一条边或者翻转路径加边。
朴素的搜索算法不断拓展当前路径。
然而每次的所有合法拓展可能有优劣之分，我们尝试贪心地在每次拓展时找到“最优”的边。
我们混合使用搜索和贪心，或者在“弹性大”的地方使用搜索确定大分支。
最后，我们可以执行算法多次，通过加入随机化元素与卡时获得更优解。
在特定的数据下，也可对不同数据采用不同的启发式算法。
因为用到了卡时，所以我们可以对算法进行一些精细的常数优化（可以发现这个算法的时间瓶颈在于判断线段是否相交，可以用快速排斥方法优化），优化后程序可以执行更多次，获得更优解。

时空复杂度:

因为卡时所以时间复杂度为时限。
空间复杂度看具体使用的启发式方法，大致为O(N+M).