集训队作业题解

镇海中学黄致焕

2016 年 1 月 11 日

Contents

1 AUG15

1.1 DISTNUM

题目大意

给定一个长度为n的数组A。现在有n个询问。

1、询问一个区间中所有三元组乘积的和。

2、修改一个元素。

3、删除一个元素。

4、插入一个元素。

5、输出区间内不同元素的个数。

题解

由于题目没有要求强制在线，所以我们可以先读入所有操作，用平衡树维护删除插入操作，给每一个点一个确定的下标，之后就变成线段树上操作了。

定义一个点的前驱为它前面第一个和他权值相等的点。

操作1：用树状数组套线段树，维护区间中每个元素三次方的和、二次方的和、一次方的和即可。

操作2、3、4：线段树上单点修改。

操作5：即询问区间内前驱在区间外的点的个数，用树状数组套线段树即可解决。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log^2 n)$。

2 JULY15

2.1 EASYEX

题目大意

有一个K面的骰子，投到每一面的概率都是完全相同的，现在你需要投n

次，设投完之后数字i出现了a_i 次。

求$\sum_{i=1}^{L} a_i^F$的期望值。

题解

假设$X_{i,j}$表示第i次是否投到j，如果是，那么值是1，否则值是0。

答案可以表示成$\sum_{i=1}^{L}(\sum_{j=1}^{n} X_{i,j})^F$。

现在将这个式子展开，我们发现每一项都包含了1 F个不同的X。

可以通过一个简单的DP来求得包含i个不同X 的系数f_i。

$f_i = i^F - \sum_{j=1}^{i-1} f_j * C_i^j$。

之后考虑DP。令$dp_{i,j}(0 \le i \le L)$为前i个数字，有j 个不同的X 的方案数。可以得到转移方程：

$dp_{i,j} = \sum_{k=1}^{min(j,F)} dp_{i-1,j-k} * f_k * C_j^k$

用FFT+快速幂加速计算即可。

复杂度

时间复杂度$O(LF \log LF \log L)$，空间复杂度$O(LF)$。

2.2 HAMILG

题目大意

两个玩家在一张n个点m条边的无向图上进行游戏，A 先选择一个起始点将硬币放在这个点上，之后两人轮流操作，B 先手，每次将硬币沿着一条边移动到另一个点上。

硬币不能重复经过同一个点，无法操作的人输掉。

现在给出这张图，求有多少个起始点可以使A胜利。

题解

考虑带花树算法，对于一个点，如果有一个最大匹配不包含这个点，那么这个点就是可以使A胜利的起始点。

因为B永远也不可能选到一个未匹配点，所以只要A每次选择B选择的点的匹配点就稳操胜券。

这样我们可以先用带花树算法求出这张图的最大匹配，之后我们可以从每一个未匹配点x开始bfs，将所有bfs到的颜色为1的点置为满足要求的点，因为只要将这个点到点x路径上的所有边翻转即可使这个点成为未匹配点。

最后统计所有满足要求的点即可。

复杂度

时间复杂度$O(n^{0.5}m)$，空间复杂度$O(m)$。

3 JUNE15

3.1 CHEFBOOK

题目大意

现在有n个人，和m条关系。

每一条关系表示第x人对第y个人喜爱值为$L_{x,y}$，要求你通过调整将喜爱值调到$S_{x,y}$到$T_{x,y}$之间。

对于每个人，定义两个值P_x,Q_x，分别表示对于所有的关系(x, y)，给$L_{x,y}$增加P_x，对于所有的关系(y, x)，给$L_{x,y}$减少Q_x。

现在要求在满足题目要求的情况下最大化$\sum L_{x,y}$。

题解

由于这道题是给定一系列线性约束并且求线性最大值，所以显然得想到了单纯形算法。

然而由于m的范围较大，单纯形的复杂度不满足要求，并且所有的变量的取值皆要求为整数，所以这里我们不使用单纯形来解决问题。

对于所有的约束，都是形如$P_y - Q_x <= S_{x,y} - L_{x,y}$ 以及$Q_x - P_y <= L_{x,y} - T_{x,y}$的。

这里如果我们将问题转化为对偶问题，就会发现每一个变量只在两条约束中出现，并且一个为+1，一个为-1，所以我们可以将变量作为边，将约束作为点，建出一张费用流。

对于一个约束，如果其约束条件为负，那么向汇连一条流量为约束条件绝对值的边。反之则由源向其连一条流量为其约束条件的值得边。对于一个变量，我们将它所在的两条约束连起来，流量为无穷，费用为要求最小化的式子中的系数。

这样我们通过跑费用流就得到了答案。

由于题目要求构造方案，所以我们考虑，如果在费用流中有一条边被流了，那么这条边所对应的变量对偶回来的约束一定被用到了，所以我们可以用差分约束求出一组解。

复杂度

时间复杂度$O(maxflow(n,m))$，空间复杂度$O(n+m)$。

4 MAY15

4.1 GRAPHCNT

题目大意

给定一张n个点m条边的有向图，统计点对$(x,y)(x<y)$ 的数量，满足从点1 到点x 和点1到点y 的路径除了在点1 没有交。

数据范围$n \le 100000, m \le 500000$

题解

考虑两个点x,y，如果从点1到这两个点的路径同时必经一个点，那么这条路径一定有交。

所以我们只要计算出每个点的最近必经点，之后在必经点树上dp 减去所有有交点对即可。

计算有向图最近必经点可以使用dominator tree 解决。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

4.2 CBAL

题目大意

一个字符串是平衡的当且仅当它的每个字符都只出现了偶数次。

现在有Q组询问，每组询问一个区间$[l, r]$ 中，所有平衡字符串长度的$TYPE$ 次方的和。$(0 \leq TYPE \leq 2)$

题解

首先我们可以用一个26位二进制数表示每种字符出现次数的奇偶性。

之后任意两个相同的数字就对应了一个平衡字符串。

考虑分块。

我们预处理任意两块之间的区间中所有平衡字符串0 次、1 次以及2 次的和。

再预处理每一块即它之前每种元素出现的位置的0次、1 次以及2 次的和。

对于一组询问，我们只要将中间的所有整块答案取出，之后再对两侧的多余元素统计即可。

复杂度

时间复杂度$O(n\sqrt{n})$，空间复杂度$O(n\sqrt{n})$。

5 APRIL15

5.1 LPARTY

题目大意

给定n个长度为m的字符串，对于字符串$s_{0..m-1}$，有$s_i = 'a' + i$ 或$'A' + i$。

对于一个字符串，它包含所有拥有它这个子集的字符串。

对于一个字符串，如果有一个他所包含的字符串不在这n 个字符串中，那么他就是不合法的。

现在你需要选出总长度最小的一些合法字符串，使得他们包含读入的所有n个字符串。

题解

由于数据范围较小，所以我们考虑使用搜索算法。

首先检索出所有合法的字符串。

对于两个合法字符串，如果其中一个包含另一个，那么那个被包含的就没有任何意义，我们可以将它删除。

由于一共只有2^n个不同的字符串，所以我们可以用一个大小小于2^{2^n}的数字来表示每个字符串是否存在，这在搜索时可以大大减小常数。

只要再加上可行性剪枝以及当前最优解剪枝即可通过。

复杂度

时间复杂度$O(2^(2^n))$，空间复杂度$O(m)$。

5.2 BWGAME

题目大意

给定一个由0,1组成的行列式，满足行列式的每一行中的1 都是连续的一段。

求这个行列式的值。

题解

考虑高斯消元算法。

我们从左向右枚举行列式的每一列，每次我们将所有以这一列为左边界的行取出来，从中选取一个右边界最左边的行。

之后将所有其他行异或这一行，也就是将前面一段减去。

减去前面的一段之后，剩下的所有行的左边界都相同，我们只要将它都挂到新得到的左边界上即可。

由于要求支持最小值和合并，所以可以用可并堆来维护。

复杂度

时间复杂度$O(n\log n)$，空间复杂度$O(n)$。

6 MARCH15

6.1 TREECNT2

题目大意

给定一棵n个节点的树，每条边有边权V_i，以及Q 次操作，每次操作修改一条边的权值。

在第一次操作前以及每次操作后输出这棵树上满足路径上所有边权最大公约数为1的数量。

题解

用f_i表示树上是i的倍数的路径条数，我们发现$ans = \sum f_i * \mu$。

所以我们对于每一个μ不为0的数x，统计出所有是它倍数的边。之后我们可以通过并查集计算出每时每刻x倍数的路径条数。

由于需要复原操作，所以需要用启发式合并的并查集方便删除。

最后我们将所有的答案合并即可。

复杂度

时间复杂度$O(V + 2^s(n + Q^2)\log n)$，空间复杂度$O(2^s n)$，其中$s$为1到$V$中最多的质因子个数。

6.2 RNG

题目大意

现在给定一个递推数列，有递推式

$A_i = (A_{i-1} * C_1 + A_{i-2} * C_2 + ... + A_{i-k} * C_k)\ mod\ 104857601$

现在给定数列前k项，$A_1, A_2, ..., A_k$和k个系数$C_1, C_2, ..., C_k$，以及n，要求输出A_n。

题解

考虑我们可以将A_i分解成$A_{i-1} * C_1 + A_{i-2} * C_2 + ... + A_{i-k} * C_k$，所以

将A_n分解成$A_1, A_2, ..., A_k$的想法就自然生成了。

由于将A_i分解的方式类似多项式中的降次，所以我们可以将下标转换为幂，写成$x^i = x^{i-1} * C_1 + x^{i-2} * C_2 + ... + x^{i-k} * C_k$。

这样我们只要将x^n降次即可。

由于在多项式中，降次的过程等同于取模，所以我们将降次方程的右侧移到左侧，将i替换成k，就将问题转化成了多项式取模：

$x^n \ mod \ (x^k - C_1 * x^{k-1} - C_2 * x^{k-2} - ... - C_k)$

我们可以用快速幂来求x^n，在乘的同时用NTT加速多项式取模，这样就可以算出x^n和$x^{k-1}, x^{k-2}, ..., 1$ 的关系，之后直接将$A_1, A_2, ..., A_k$带入，即可得到答案。

复杂度

时间复杂度$O(k \log k \log n)$，空间复杂度$O(k)$。

7 FEB15

7.1 DEVLOCK

题目大意

求各数位和小于等于k的被p整除的m位数的个数（可以有前导0）。

题解

考虑先预处理模p余每种余数的位数。

之后对每一种余数计算出各位和为i时的方案数。

设某种余数位数为s，各位和为i的方案数的母函数为g(x)。

这样我们可以得到$g(x) = f(x)^s$。

其中$f(x) = x^9 + x^8 + ...1$表示一位的情况。

这一部分我们可以用FFT+快速幂解决。

之后考虑将每一部分合并起来。

我们可以用一个二维数组h[i][j]表示余数为i各位和为j 的方案数。

之后我们可以将g数组拆分，然后分别和h数组相乘，再将所得存到h 数组中即可。

这里注意，我们需要先将h数组和拆分后的g数组全都DFT，之后等乘完再IDFT 回来。

复杂度

时间复杂度$O(mm\log mm\log n + p^3 * mm + p^2 * mm\log mm)$，空间复杂度$O(p * mm)$。

7.2 CUSTPRIM

题目大意

定义三元组(a, b, c)的乘法运算，其中$c = 11$ 或者$c = 24$。

如果令(a_1, b_1, c_1)和(a_2, b_2, c_2)相乘，令

$s = (a_1a_2 + b_1b_2 + c_1c_2) + (a_1b_2 + b_1c_2) + (c_1 + c_2), t = \lfloor \frac{s}{2} \rfloor + 16(c_1 + c_2) - c_1c_2$

$A = (t - 2(a_1b_2 + b_1a_2) - (a_1c_2 - c_1a_2) + 33(a_1 + a_2) + (b_1b_2 - a_1a_2))$

$B = (t - 5(a_1b_2 + b_1a_2) - (c_1b_2 - b_1c_2) + 33(b_1 + b_2) + (2b_1b_2 + 4a_1a_2))$

如果s是偶数，那么结果是$(A - 540, B - 540, 24)$，否则结果是$(A - 533, B - 533, 11)$。

定义单元A是对任意B都满足$A * B = B$的三元组，定义$seroA$ 是对任意B都满足$A * B = A$ 的三元组。定义一个三元组是素数当且仅当这个三元组不能被表示成两个非零非单位元的三元组的积。

给定一个三元组，判断它是不是素数。

题解

令ω为满足方程$\omega^2 = \omega - 3$的解，即$\omega = \frac{1+\sqrt{-11}}{2}$，那么问题就转化成了判断域$Z[\omega]$下数$a + b\omega$是否为素数。

定义共轭$(a + b\omega)' = (a + b - b\omega)$，那么令$Nx = xx'$，那么就有如下结论:

1、如果x不是整数，那么x是质数当且仅当Nx是质数。

2、如果x是整数，那么x是质数当且仅当x 是质数且要么$|x| = 2$，要么$|x| \neq 11$ 且-11在模x的域下没有二次剩余。

所以用miller rabin和欧拉判别法即可解决问题。

复杂度

时间复杂度$O(T\log|a|)$，空间复杂度$O(1)$。

8 JAN15

8.1 XRORS

题目大意

给定一个序列a，要求支持以下5种操作。

1、在末尾添加一个元素x。

2、在区间$[L,R]$中找到一个元素y，最大化x^y。

3、删除最后k个元素。

4、在区间$[L,R]$中找到一个元素小于等于x。

5、在区间$[L,R]$中找到第K小的元素。

题解

由于只有在序列末尾的修改操作，所以我们可以考虑建立函数式trie 树维护。

trie树的每个节点有两个子节点，分别表示这一位是0 或1。

这样操作1和操作3只要对最后的一系列root进行操作即可。

操作2、4、5我们都可以通过在trie树上从根向下走来确定满足条件的答案。

复杂度

时间复杂度$O(n\log n)$，空间复杂度$O(n\log n)$。

8.2 RANKA

题目大意

给定一张9*9的围棋棋谱，黑白两方分别操作，每次一方可以选择不下或者下子。

要求你同时操作两人，保证不能重复出现同一种局面：棋谱上所有子和当前操作者同时相同。

题解

由于一个人可以选择不下或者下子，所以我们不应该从普通的围棋角度考虑这道题。

比较简单的一个做法是第一个人不断地从左上角向右上角下子，直到只剩下一个空格，这时后手下一个子在那个空格上。

之后整个棋盘只剩下一个黑子，所以后手可以重复先手这一过程，之后先手再填上最后一个空格。

这样最多可以进行81轮，每一轮可以走161步，总共超过了要求的10000步，所以满足要求。

复杂度

时间复杂度$O(n)$，空间复杂度$O(1)$。

8.3 SEAND2

题目大意

有一个整数A，在十进制下不含0。有n个整数，$B_1, B_2, ..., B_n$。

定义一个关于A的函数f：$f(A) = \sum_{i=1}^{n} A \ mod \ B_i$。

现在请你将A中的数字进行重排，最小化$f(A)$的值。

题解

首先对于$10^0, 10^1, ..., 10^{n-1}$预处理对$B_1, B_2, ..., B_n$ 分别取模的值。

这样我们就可以在$O(n)$的时间内求出交换两位对答案的贡献了。

之后用模拟退火算法，每次随机两位根据贡献随机是否交换，即可通过。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n \log n)$。

9 DEC14

9.1 DIVIDEN

题目大意

给定一个n度的角，请你用尺规作图将它分成n个1度角。

你可以使用以下两个操作：

1、在两个点之间连一条直线。

2、以一个点为圆心，画一个半径为两个点之间距离的圆。

题解

由于我们可以通过画五边形构造出一个36度角，同时可以通过平分一个60度角构造出一个30 度角，这样我们显然可以构出一个3 度角。

构造出一个3度角之后，我们显然可以将所有模3 不余0 的所有角都平分，这样问题就解决了。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

9.2 RIN

题目大意

有n门课程和m个学期，每一门课程都必须选择一个学期来学习。

一些课程有前置课程。

在第i个学期学习第j门课程可以获得$b_{i,j}$的分数，如果$b_{i,j}$ 为-1，表示第i个学期不开设j 课程。

求最多能拿到多少分。

题解

考虑将问题转化成最小割。这里设S为所有分数中的最大值。

首先对于第i门课程，拆成$m+1$个点$a_{i,0}..a_{i,m}$。

在点$a_{i,j}$到点$a_{i,j+1}$之间连一条流量为$S-b_{i,j+1}$ 的边(如果$b_{i,j+1}=-1$ 就连一条流量为无穷大的边)。

之后对于课程x和课程x的前置课程y，对于$0 \le j < m$，在点$a_{y,j}$ 到点$a_{x,j+1}$之间连一条流量为无穷大的边。

最后用最大流计算出这张图的最小割，用$S*n$ 减去之即为答案。

复杂度

时间复杂度$O(maxflow(n,nm))$，空间复杂度$O(nm)$。

9.3 KALKI

题目大意

给定二维平面上n个点，要求返回一棵生成树，使得最大点的点权最小。

其中点权的定义为：对于一个每一个点，对于连接它的每一条边，找到一条最长的，之后将所有和该点距离小于等于那条边长的点的点权增加1。

题解

仔细读题，发现在图是最小生成树时的解就较优，所以我们使用prim算法求出该图的最小生成树就可以通过该题。

复杂度

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

10 NOV14

10.1 FNCS

题目大意

有一个长度为n的数列a，定义m个函数，每个函数包含两个值l_i,r_i。

函数i的值定义为数列a中下标l_i到r_i 的元素的和。

现在给定这m个函数，要求支持以下操作：

1、修改数列a中的元素。

2、询问第L个到第R个函数值的和。

题解

考虑分块的做法。

将函数按照顺序从前往后分成K块，其中对于每一块求出对于数组a 中的每一个元素，这一块中被计算了几次。

对于操作1，我们只要对于所有K个块进行修改即可。这部分复杂度为$O(K)$

对于操作2，我们对于其中所有整块直接取答案，对于边缘的一些函数用树状数组求值。这部分复杂度为$O(m \log n/K)$。

所以这里取K值为$\sqrt{m \log n}$

复杂度

时间复杂度$O(n\sqrt{m \log n})$，空间复杂度$O(n\sqrt{m \log n})$。

10.2 SEAORD

题目大意

有n个程序，每个程序都需要在两台机器上分别运行。第i 个程序需要在第一台机器上跑A_i的时间，在第二台机器上跑B_i的时间。

一个程序不能同时在两台机器上跑，请问最少要多少时间跑出所有程序。

题解

首先定义$Sa = \sum_{i=1}^{n} A_i$，$Sb = \sum_{i=1}^{n} B_i$

那么可以得到$ans = max(Sa, Sb, max(A_i + B_i))$。

由于第三种情况的构造显然，这里只讨论前两种。

因为直接做比较麻烦，我们可以考虑随机化。

给所有程序一个随机的顺序，之后从前往后将每个程序的A 部分按顺序从前向后做，将B 部分按顺序从后向前做，直到出现冲突。

出现冲突后就将剩下的程序在AB中分别按顺序从前往后摆放，再判定一下合法性即可。

由于冲突的情况概率较少，所以一般情况下多做几次一定可以求出答案。

复杂度

时间复杂度$O(n*$玄学$)$，空间复杂度$O(n)$。

11 OCT14

11.1 TRIPS

题目大意

有一棵n个节点的树，其中每条边的边权为1 或2。

现在有Q组询问，每组询问表示从点x走到点y，一次最多走z 的长度并且每次必须都在节点上结束，需要走几次。

题解

首先对于一组询问，我们可以将它拆成从两侧走到中间来合并答案。

对于$z \geq \sqrt{n}$的询问，有$ans \leq \sqrt{n}$。

这时我们可以预处理每个节点的第2^k个祖先，用倍增算法一次一次向上走。

对于$z < \sqrt{n}$的询问，一共只有$\sqrt{n}$种不同的z 值，

对于所有z值相同的询问，我们可以预处理每个节点向上走2^k 次走到的点，之后用倍增解决问题。

复杂度

时间复杂度$O(n\sqrt{n}\log n)$，空间复杂度$O(n\sqrt{n}\log n)$。

11.2 BTREE

题目大意

有一颗n个节点的树，树边权均为1。

有Q组询问，每组询问给定K个警卫，每个警卫在树上的一个点上。

对于第i个警卫，他在第A_i个点，可以管辖距离A_i 点$\leq B_i$的点。

现在求对于一组询问，所有的警卫可以管辖多少不同的点。

题解

首先我们进行点分治，对于每个子树，我们处理出所有点到重心的距离，之后用一个数组对于每个距离保存这个距离的点的个数。

这样我们就可以在$O(\log n)$的时间内求出到一个点距离$\leq X$ 的点。

对于一组询问，我们建立一棵虚树来处理。

考虑两个相邻的警卫i, j，如果它们没有互相包含，那么他们同时包含的节点可以用距离它们之间的某个点一定距离的点来表示。

这样我们只要在虚树上进行dfs计算答案即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log n)$。

12 SEPT14

12.1 QRECT

题目大意

请你维护以下3个操作

1、I $x1$ $y1$ $x2$ $y2$添加一个左下角为$(x1, y1)$，右上角为$(x2, y2)$的矩形。

2、D x将添加的第x个矩形删除。

3、Q $x1$ $y1$ $x2$ $y2$询问左下角为$(x1, y1)$，右上角为$(x2, y2)$的矩形与几个已添加的矩形相交。

题解

由于题目没有要求强制在线，所以我们可以考虑CDQ分治来解决。

在分治的时候，我们只要对于分治右边的每个询问统计出和他不相交的矩形数再用总数减去即可。

对于分治左边的矩形，我们可以用扫描线的思想用树状数组来维护并进行统计。

删除操作只要计算负贡献即可解决。

复杂度

时间复杂度$O(n \log Q)$，空间复杂度$O(n)$。

12.2 FIBTREE

题目大意

给定一棵n个节点的树，要求支持以下4种操作。

1、给一条链加上斐波那契数列。

2、询问以点x为根时以y为根的子树的权值和。

3、询问路径权值和。

4、回到第x个操作后的状态。

题解

我们可以用树链剖分来维护点权，这样就可以处理链上修改、子树询问和链上询问问题了。

对于斐波那契数列，我们可以用斐波那契数列的通项公式，将两项拆开计算。我们只要求出两项在模的意义下的值，就将问题转换成了区间加等比数列问题。

区间加等比数列是经典问题，可以用线段树打标记简单解决。

由于有4操作，所以我们只要使用可持久化线段树来维护树链剖分即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log^2 n)$。

13 AUG14

13.1 SIGFIB

题目大意

给定n,m。

求$\sum(6*x*y*z*fibo_x*fibo_y*fibo_z)\ mod\ m, x+y+z=n$

题解

通过计算发现这个数列的生成函数为$f(x)=6g(x)x^3(x^2+1)^3$。

由此可得$[x^n]g(x)=\frac{1}{3125}[25C_{n+5}^5(fib_{n+6}+2fib_{n+5}+150C_{n+4}^4fib_{n+5}+5C_7^2C_{n+3}^3(fib_{n+4}+2fib_{n+3})+5C_8^3C_{n+2}^2fib_{n+3}+C_9^4(n+1)(fib_{n+2}+2fib_{n+1})+C_{10}^5fib_{n+1}$

之后用矩阵乘法计算出fib_i后直接计算答案即可。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

13.2 PUSHFLOW

题目大意

给定一张n个点m条边的仙人掌，每条边有边权。

现在有Q组询问和修改，对于每一组询问，询问两个点之间的最大流。

题解

首先用tarjan将仙人掌上的环都缩起来。

缩环后，对于一个环，我们将它连向它父亲的那个点成为top点。

之后将缩环后的树进行树链剖分，对于一个环的重儿子，我们随时维护它走到top的情况。

同时再用一个线段树维护每个环上边的情况，这样在走轻边时，我们就可以通过线段树上两段区间的查询来求出环上两条路上的最小值。

对于修改，我们先判断修改的边是否在环上，如果在环上就将保存环上边的线段树上修改，之后再修改该环重儿子走到top的情况。如果不在环上，那就直接在树链剖分处修改该边的权值即可。

对于询问，只要判一下两个点是否在两个点lca的环上，根据不同的情况讨论即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n)$。

14 JULY14

14.1 GNUM

题目大意

给定两个长度为n的数列a,b，求最大匹配$(i,j)->(k,l)$ 个数，满足$a_i < b_j, a_k > b_l$ 并且$gcd(a_i,b_j,a_k,b_l) > 1$。

题解

由于是一道经典的最大匹配问题，显然地想到了匈牙利算法。

但是由于数据范围过大，不能使用匈牙利，所以我们考虑将一些边整合起来，用网络流解决。

首先处理出数列a,b中包含的所有质数，之后对于每个质数将所有同时是他的倍数的点对处理出来。

之后对于每个质数建点，连接那些点对，最后跑一遍网络流即可得到答案。

复杂度

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

14.2 SEAEQ

题目大意

定义两个数列A, B是相似的，满足$|A| = |B|$并且对于任意的i 满足A_i在A数组中的排名等于B_i在B数组中的排名。

定义函数$F(A, B)$表示满足$A[l..r]$相似于$B[l..r]$，并且$A[l..r]$中的逆序对数小于等于E的点对(l, r)的个数。

求$\sigma F(A, B)$，其中A和B取遍长度为n的全排列。

题解

首先DP预处理出长度为i的逆序对数小于等于E的序列个数。

状态转移方程为

$$F[i][E] = F[i][E-1] + \sum_{j=max(E-i+1,0)}^{E} F[i-1][j]$$

其中求和部分可以使用前缀和维护。

之后考虑枚举长度，长度为i的序列的贡献为

$$F[i][E] * (n-i+1) * C_n^i * (n-i+1)!$$

最终得到

$$ans = \sum_{i=1}^{n} F[i][E] * (n - i + 1) * C_n^i * (n - i + 1)!$$

复杂度

时间复杂度$O(n * E)$，空间复杂度$O(n * E)$。

14.3 GERALD09

题目大意

现在有一个$n * m$的空矩阵，要求给每一位填上A, C, G, T 中的一个字母，使得该矩阵的不同子矩阵个数最接近K。

题解

首先，我们随机生成一个初始矩阵，之后二分需要置0的格子数，得出一个较接近答案的初始矩阵。

之后我们每次随机一个格子，随机修改它的权值并查询答案，如果比之前的解更优就采取它。

这里我们要在$n^2 * m^2$的时间内求出答案。我们可以给每个单位矩阵一个$hash$值，对于一个子矩阵，我们只要将去掉最后一行的矩阵的$hash$值和最下面一行矩阵的$hash$ 值合并即可。

卡时进行T次随机即可通过。

复杂度

时间复杂度$O(T * n^2 * m^2)$，空间复杂度$O(n * m)$。

15 JUNE14

15.1 SEAARC

题目大意

给定一个数列a_i，统计满足$a_i = a_k, a_j = a_l, a_i \neq a_j, i < j < k < l$的不同点对$(i, j, k, l)$个数。

题解

将每种a的值按照出现次数分成两类，A类包含所有出现次数较大的值，B类包含其他值，设置这个分界线为x。

对于每个A类值，只要将所有的点扫一遍，即可得到A类和其他所有的答案和，复杂度为$O(n^2/x)$。

对于所有B类之间的答案，我们只要扫一遍所有的B类点，暴力使用树状数组维护即可，复杂度为$O(x * n \log n)$。

由此，只要将x设定为$\sqrt{n/\log n}$即可达到最低复杂度。

复杂度

时间复杂度$O(n\sqrt{n \log n})$，空间复杂度$O(n \log n)$。

15.2 TWOCOMP

题目大意

给定一棵树和两种类别分别$m1, m2$条链，每条链都由一个权值，要求一个链的集合，满足其中没有两条不同类别的链相交。

要求最大化集合中链的权值和。

题解

首先可以用lca预处理出每两条链是否相交，之后建出二分图：

1、在两个相交的链之间连上流量为无穷大的边。

2、源向第一种链连该链权值流量的边。

3、第二种链向汇连该链权值流量的边。

之后直接跑网络流即可。

复杂度

时间复杂度$O(maxflow(m1 + m2))$，空间复杂度$O(m1 * m2)$。

16 MAY14

16.1 ANUDTQ

题目大意

给定一棵n个节点的有根树，m个操作或询问。

有4种不同的操作或询问

1、为一个节点添加一个儿子节点；

2、将一个子树中所有点权增加一个值；

3、删除一个子树的点；

4、询问一个子树中的点权和；

题解

在解决静态树问题的时候，维护子树信息我们常常使用dfs 序来解决。

在有根树上，一个子树就对应该树的dfs序上的一个区间。

由于题目要求修改，修改后子树所对应的区间会发生变化，再对应时会有些麻烦。

因此，我们使用和dfs序类似的——括号序列来解决这个问题。

括号序列即在dfs时，访问和离开一个点时都记录的序列。

显然的，括号序列上以一个点为根的子树所对应的区间一定是这个点的左右括号中间的那一段。

这样我们可以很方便的将子树对应到区间上，只要在修改的时候维护括号序列即可。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

16.2 SEINC

题目大意

给定两个长度为n的数列a,b，一次操作可以将数列a中一个区间中的元素，

增加1 之后模4。

问最少操作几次可以将数列a转变成数列b。

数据范围$n \le 100000$，数据组数$T \le 10$。

题解

由于从数列a变换到数列b，我们可以将两个数列相减得到数列c，表示所有操作需要改变的地方。

如果操作时只有增加1而没有取模，那么这题的答案便是

$$\sum_{i=1}^{n} c_i - c_{i-1}(c_i > c_{i-1})$$

证明可以参考noip2013“积木大赛”一题。

由于题目有取模，所以我们可以给任意一项增加4或者减少4，只要满足每一项都大于0 即可。

所以我们可以定义d数组$d_i = c_i - c_{i-1}$，之后通过修改d 数组的值来优化答案。

由于c数组中的每一项都必须非负，所以d数组的前缀和必须非负。

每次我们可以将d数组中的一项增加4或者减少4，之后求d 数组中的所有正项和。

通过简单的观察、试验，我们可以发现：

对于非负项，如果我们给他增加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，答案并不会改变，显然也不会更优。

因为d数组中有加有减较为麻烦，所以我们可以先将所有项都加到非负，之后只考虑减少。

根据d数组中每一项的值，我们可以求出给它减少4对答案的贡献。

所以问题就转化成了，在第i项之前有$d_i/4$ 项可以选择，已知每一项的贡献，要求最大化贡献。

对于这个经典问题，我们只要将贡献排序之后从大到小做即可。

由于题目权值只有3种，所以我们可以用基数排序在$O(n)$ 的时间复杂度下完成排序。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

16.3 ANUMFS

题目大意

有一张地图，由n个边界点描述。

其中有一个关键点（也可能不在地图中），你需要找出它的位置。

有m种飞机，每种飞机有两个值R, C，表示他所能探索的范围和它的费用。

每次操作可以将一架飞机派向(x, y)，需要支付$C * (x + y)$ 的费用。

如果关键点和飞机的哈夫曼距离小于等于R，那么就能探索到关键点，否则就不行。

要求确定关键点的位置并最小化费用。

题解

如果当前有N个待确定点，加入一次操作圈定了X个点，那么期望剩下的待确定点个数为$((N - X) * (N - X) + X * X)/(N * N)$。

由于要最小化费用，所以要尽量减少询问的次数，也就是尽量使待确定点个数更少。

在X=N/2的时候，期望值最小。

由于要求飞机派往地图中的点上，所以依靠大半径飞机不断划分剩余区域的思路就会遇到阻碍。

基于此，我们可以使用随机化来近似地解决这个问题。

由于要采用随机化，所以一个可靠的股价函数是必不可缺的。

因为同时要求花费最小并且剩余个数最少，所以这个股价函数必须和两者同时相关。

简单的，我们可以直接用他们的积作为估价函数即可。

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，再派一架R为0的飞机探索该点就好了。

17 APRIL14

17.1 GERALD08

题目大意

给定一棵n个节点的树，其中每条边是黑色或白色。

两个人轮流操作，一个人只能割黑边，一个人只能割白边。如果一个人无法操作，那么他就输了。

现在请问两个人分别先手时谁会胜利。

题解

对于一个局面，我们定义一个函数f，表示这个局面对双方的有利程度。

对于一个空节点，它的函数值为0。

一个节点的函数值为所有孩子的函数贡献之和。

对于一个节点，如果连接它的是黑边，那么它对父亲函数值的贡献为$\frac{f+a}{2^{a-1}}$(其中a 为满足$f+a>1$ 的最小正整数。

如果连接它的是白边，那么它对父亲函数值的贡献为$\frac{f-a}{2^{a-1}}$(其中a 为满足$f-a<-1$的最小正整数。

如果根节点函数值为0，那么后手必胜，若根节点函数值为正，那么割黑边者必胜，若根节点函数值为负，那么割白边者必胜。

由于本题精度要求较高，所以我们可以用高精度来实现，用启发式合并维护小数部分即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log n)$。

18 MARCH14

18.1 GERALD07

题目大意

给定n个点，m条边，q组询问，询问编号在l到r 之间的所有边组成的无向图的联通块个数。

题解

考虑link-cut-tree，对于一条边i，我们要预处理出最大的f_i 满足当第f_i 条边到第$i-1$条边同时存在时，边i 的两端在同一联通块中。

预处理出f_i之后，对于区间$[l,r]$的询问，我们只要统计出在区间$[l,r]$ 中，满足$f_i < l$ 的边数，即边区间$[l,r]$ 组成的生成森林中的边数。用总点数n 减去，即可以得到联通块数。

由于以上问题是一个二维数点问题，所以可以使用函数式线段树解决。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n \log n)$。

18.2 STREETTA

题目大意

给定一个数列，要求维护两种权值的和。

有三种操作：

1、将数列1的区间[l,r]和一个等差数列求max。

2、将数列2的区间[l,r]加上一个等差数列。

3、求两个数列第i位权值的和。

题解

我们可以用线段树来维护。

操作1：我们在修改线段树上的一个节点时，我们可以将该点的标记和当前加入的等差数列进行比较，将其中控制较小区间的那个传入线段树的子节点。

操作2：在线段树上直接打标记不下传即可。

操作3：在两个线段树中分别统计答案，之后将其加起来即可。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

19 FEB14

19.1 DAGCH

题目大意

给定n个点的有向图，点按有向图dfs的顺序编号。

对于每个点x，求一个编号最小的点y，可以通过只走满足编号¿x 的点走到x。

最后输出每个点y被多少个点x求得。

题解

由于点按照有向图dfs顺序编号。

所以对于一条边$x->y$，如果$y>x$，那么y一定是x的子孙。

由此我们可以得到，对于点x满足条件的路径必须不能中途经过点x 的祖先，而点y 也一定是点x 的祖先。

所以这个点y就是点x的最近半必经点。可以使用dominator tree 算法解决。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

19.2 COT5

题目大意

要求维护一个treap，有以下3种操作。

1、插入一个关键字为k，权值为w的点。

2、删除一个关键字为k的点。

3、返回关键字为ku和kv两个节点之间的距离。

保证任意时刻树中节点key和weight都是两两不同的。

题解

对于操作3，我们可以处理出ku和kv的lca，之后通过三个点的深度来确定两点间的距离。

显然的，将关键字排序后，ku和kv的lca就是他们之间权值最大的点。

所以我们只要处理一个节点的深度即可。

考虑先将所有操作读入，之后离散化，添加删除操作就变成了线段树上的修改。

定义线段树上的一个函数get(a,b)表示对于线段树上的一节点a 中，权值为b 的点的深度。

对于每一个节点，时时维护$get(leftson'skey, rightson)$ 以及$get(rightson'skey, leftson)$即可。

由于在get函数在调用时只需要向一个方向走即可，所以一次get 函数的复杂度为$\log n$。

一次询问需要调用$\log n$次get函数，

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log n)$。

20 JAN14

20.1 CNTDSETS

题目大意

在n维空间上有一些点组成一个点集，定义两个点之间的距离等于坐标差最大的维度的坐标差。

如果两个点集互相可以通过平移得到，那么他们等价。

求两两不等价的点集的个数满足这个点集中最远的两个点距离为d。

题解

我们假设所有点的每一维坐标都在$[0, d]$这个范围内。

由于要去重，所以我们假设每一维0上必须都有点。

因为距离必须是d，所以必须至少有一维在d 上有点。

考虑容斥。

f_i表示前i维0上有点，后面$n-i$维0上没点，并且有一维在d 上有点的方案数。

然后我们可以得到

$$f_i = 2^{(d+1)^i * d^{n-i}} - 2^{d^i * (d-1)^{n-i}} - \sum_0^{j<i} C_i^j * f_j$$

最后根据这个式子递推，答案就是f_n。

复杂度

时间复杂度$O(n^2)$，空间复杂度$O(n)$。

20.2 TAPAIR

题目大意

给定一张无向联通图，现在问有多少种方案可以选择两条不同的边删去并使得这张图依然联通。

题解

首先将这张图的dfs树建出，剩下返祖边和树边。

考虑删的两种边的种类。

1、一条树边和随便的另外一条边。只要这条树边没有被任何返祖边覆盖，那么删了他本身之后整张图就不连通了，所以只要随便再删一条就好了。

2、两条返祖边。显然是合法的，因为树边本身是联通的。

3、一条树边一条返祖边。如果有且只有一条返祖边从他的儿子导向他的祖先，那么这两条边就是不合法的，算入答案。只要dfs一遍即可统计完成。

4、两条树边。如果这两条树边被完全相同的返祖边覆盖，并且中间的所有点不能通过他们其他的儿子返祖，那么这两条边也是不合法的，算入答案。具体实现可以用主席树解决。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n \log n)$。

21 DEC13

21.1 QTREE6

题目大意

给定一棵树，每个点有0、1两种权值。

有两种操作：

1、改变一个点的权值。

2、求一个点所在的同权值联通块中的点数。

题解

首先用树状数组维护括号序列来记录链上的前缀信息。

这样我们就可以通过二分求得一个点的祖先中深度最大的和他不同颜色的点。

之后我们考虑维护以一个点为根的子树中与这个点同权值联通块中的点数。

对于修改操作，我们只要将它以上的所有同颜色点都加上它的权值并翻转颜色即可。

对于询问，我们只要找到和他相同颜色联通的深度最小点，再输出其子树中的答案即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n\log n)$。

21.2 REALSET

题目大意

给定一个长度为n的整数序列A，问是否存在一个不全为0 的整数序列B满足对于所有的$0 \le j < n$都有 $sum_{i=0}^{n-1} A_i * B_{(i+j)\ mod\ n}$。

题解

问题等价于判断数列A的循环矩阵X是否满秩。

如果由数列A构成的生成函数$f(x)$满足$gcd(f(x), x^n - 1)$ 的次数为d，那么这个矩阵的秩就是$n - d$。

利用分圆多项式，我们可以将问题转化为$f(x) * \prod_{i|d}(x^{d/i} - 1)$被$x^d - 1$ 整除(其中i 是质数)。

之后使用模拟来判定即可。

复杂度

时间复杂度$O(n\sqrt{n})$，空间复杂度$O(n)$。

22 NOV13

22.1 MONOPLOY

题目大意

给定一棵树，每个节点有一个颜色。

点u权值的表示从点u走到根的路径上经过的不同的颜色数的平均值。

定义一个函数$f(u)$表示以u为根的子树中，每个点的权值的平均值。

要求支持将一个点到根路径上所有点的颜色变为一种新颜色的操作，以及对一个点的f值的询问。

题解

使用树链剖分维护。

对于一个点，如果它和父亲颜色不同，那么它所在的子树中所有点的权值都增加1。

这里我们默认一个轻儿子和它的父亲颜色不同，重儿子和他的父亲颜色相同。

称一个点为特殊点，表示它和它的重儿子颜色不同。

我们在修改操作时，每次将重链上的特殊点全部修改，同时在走轻边时修改父亲的重儿子为特殊点。

这样一次操作为增加$\log n$个特殊点，总共$n \log n$ 个。

对于询问操作，我们可以利用树链剖分的dfs序性质，将子树对应到区间上，在线段树上区间询问即可。

复杂度

时间复杂度$O(n \log^2 n)$，空间复杂度$O(n)$。

22.2 QPOINT

题目大意

在二维平面上，给定n个国家，每个国家都是一个简单多边形。

有Q组询问，每组询问给定一个坐标，问这个坐标在哪个点内部。

强制在线。

题解

如果没有强制在线，对于所有询问和多边形的边按照y值排序，之后我们可以用平衡树维护扫描线，在平衡树中对于每一条边记录它是一个多边形的左边界还是右边界，同时记录其所在的多边形。

对于同一个y值，我们先删边再加边，这样可以避免一些比较问题，只是要注意特判横向边。

对于一组询问，我们只要找到它左侧的第一条边，判断其是右边界还是左边界，如果是左边界，那么点就在多边形内，反之相反。

由于题目要求强制在线，所以我们可以将平衡树可持久化，对于每组询问在可持久化平衡树上查询即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n)$。

23 OCT13

23.1 FN

题目大意

F_i表示斐波那契数列第i项。

已知C,P，有$F_n\equiv C(\text{mod }P)$，求$n$的值。

题解

考虑斐波那契数列通项公式$F_i=1/\sqrt{5}[(\frac{1+\sqrt{5}}{2})^n-(\frac{1-\sqrt{5}}{2})^n]$。

设$y=\frac{1+\sqrt{5}}{2}$，我们可以得到$\sqrt{5}F_i=y^n-(-\frac{1}{y})^n$。

设$z=y^n$，得到

$$\sqrt{5}F_i*z=z^2+1(n\text{ mod }2=1)$$

$$\sqrt{5}F_i * z = z^2 - 1 (n \bmod 2 = 0)$$

之后我们可以利用二次方程求根公式解出z的值，再用BSGS还原出y的值即可。

复杂度

时间复杂度$O(\sqrt{P})$，空间复杂度$O(\sqrt{P})$。

24 SEPT13

24.1 TWOROADS

题目大意

给定一张n个点的图，求两条直线，最小化所有节点到这两条线上最近的路径的平方和。保证不存在重点和三点共线。

题解

考虑这两条直线控制的点。

显然的，我们发现这两条直线夹角的两条角平分线将平面分成4 个不同的区域。

由于两条直线垂直，所以我们可以通过枚举第一条直线经过的两个点以及第二条直线经过的一个点，这样我们可以确定4 个区域。

在确定了每条直线控制的点之后，利用距离公式$\frac{|Ax+By+C|}{\sqrt{A^2+B^2}}$，通过判别式简单推导之后，我们就可以得到最小值关于$\sum x, \sum y, \sum x^2, \sum y^2, \sum xy$ 以及包含的点数的关系。

我们可以将所有点排序之后一个一个扫过来同时维护以上6 个值，计算出每一种情况的解，最后取最小值即可。

复杂度

时间复杂度$O(n^3 \log n)$，空间复杂度$O(n)$。

24.2 TMP01

题目大意

要求支持以下3个字符串操作：

1、在末尾增加一个字母。

2、在开头删去一个字母。

3、询问整个字符串的不同子串个数。

题解

使用Ukkonen算法构造后缀树可以支持1、3两个操作。

考虑2操作在后缀树上的情况，就是删去一个后缀。

所以我们只要在后缀树上暴力删除这个后缀并合并所有度数为1的祖先节点，最后更新答案即可。

删除的时候要注意将活动节点展开。

复杂度

时间复杂度$O(n^3 \log n)$，空间复杂度$O(n)$。

25 AUG13

25.1 LYRC

题目大意

给定W个词P，再给出N个句子S，求这W个词分别在所有句子中出现的总次数。

题解

首先把所有单词建成一个AC自动机，之后把每个句子分别在AC 自动机上跑一边，记录每个点的访问次数。

最后对于每个单词的接受节点，统计一下他所在的fail 树上所有节点的访问次数和即为答案。

复杂度

时间复杂度$O(W|P|+N|S|)$，空间复杂度$O(W|P|+N|S|)$。

25.2 PRIMEDST

题目大意

给定一个n个节点的树，求树上长度为质数的路径数占总路径数的百分比。

题解

首先用线性筛预处理出所有的质数。

之后考虑点分治。

在分治一个点的时候，只要用dfs对每个子树统计出连向该点的所有路径长度。

之后只要把每个子树和他之前的所有子树合并即可。

考虑如何合并。在统计出每种长度出现的次数之后，两个子树的合并就是一个卷积。

所以可以使用FFT来加速卷积。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n)$。

26 JULY13

26.1 RIVPILE

题目大意

有一条宽度为W的河，河上有N根柱子，第i根柱子坐标为X_i, Y_i，现在有M种不同规格的圆盘。对于一个圆盘，用两个值R_i, C_i来描述，分别表示半径和价格。

现在要求你购买一些圆盘放到柱子上，使得人可以过河。

题解

如果有两个圆盘满足$R_i > R_j$并且$C_i < C_j$，那么显然的，j没有存在的意义，所以我们可以将其删去。

之后我们将剩下的m种圆盘按照R的大小从小到大排序。

将每个柱子拆成m个点，每个点对应一种圆盘。从小的圆盘向大的圆盘连边权为价格差的边，从大的向小的连边权为0的边，并对可以和河两岸连接的圆盘建边。

之后对于两根柱子，我们枚举第一根柱子的每一个圆盘，向第二个圆盘最小的能与之相交的圆盘的对应点连边。

这样图就建完了，跑dijkstra算法即可。

复杂度

时间复杂度$O(NM \log NM + N^2M)$，空间复杂度$O(N^2M)$。

27 JUNE13

27.1 TKCONVEX

题目大意

有n根棍子，求是否能选出2*K根组成两个凸多边形。使得每个正好都由K根组成，且任意两边不共线。

题解

考虑只有一个多边形的情况。

由于要最小化最大边的长度并最大化其他边的长度和，所以最优情况就是将所有边排序后取相邻的K条边。

在两个多边形时，有两种情况。

1、两个多边形的边集不相交。

2、两个多边形的边集相交。

对于第一种情况，我们只要扫一遍判断每个边集是否可取即可。

对于第二种情况，我们可以枚举相交的边集（一定是相邻的2*K 条边），之后用搜索判断是否可行即可。

复杂度

时间复杂度$O(n*C_K^2)$，空间复杂度$O(n)$。

27.2 SPMATRIX

题目大意

对于一个n*n的矩阵，我们称它为特殊的当且仅当他满足以下条件：

1、$A_{i,i}=0$

2、$A_{i,j}=A_{j,i}>0(i<j)$

3、$A_{i,j}\le max(A_{i,k},A_{k,j})$

4、$0\le A_{i,j}\le n-1$

5、对于0 $n-1$中的任意一个值，存在$A_{i,j}$与之相等。

题解

可以知道答案等于$\frac{n!*(n-1)!}{3*2^{n-1}}(\frac{3n}{2}-2-\sum_{i=1}^{n-1}\frac{1}{i})$。

对于这个公式，我们只要对于每一个n预处理出答案，之后对于每个询问回答即可。

由于时限较小，所以需要尽量减少乘法和取模的次数来进行常数优化。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

27.3 CHAORNOT

题目大意

有n个数$a_1,a_2,...,a_n$，现在求从这n个数中选出最多的数，满足没有任意三个数成等差数列。

题解

考虑随机化。

我们给这n个数一个随机的顺序，之后一个一个取，如果合法就将它加入集合，否则就不加。

每加入一个数，就将它和所有在集合中的数求一下等差数列，将所有不合法的数用一个数组存下来，这样就可以$O(1)$ 判断一个数是否合法了。

假设一共选了m个数，这样一次的复杂度就是$O(m^2+n)$，卡一下时即可AC。

复杂度

时间复杂度$O(T(m^2+n))$，空间复杂度$O(n)$，其中T 为进行的次数。

28 MAY13

28.1 QTREE

题目大意

给定一个n个节点的环加外向树，每条边有一个边权，定义两点间的路径为两点间的最短路。

现在给定Q个操作，有两种操作：

1、对u到v的路径上所有边的权值取相反数

2、询问u到v的路径上一个连续的边集，使集合中所有边权和最大，输出这个最大值。

题解

对于链上的问题，我们可以用线段树维护每一段区间的最大权连续边集，最大权前缀，最大权后缀，以及最小权连续边集，最小权前缀，最大权前缀，推一下翻转标记即可。

现在问题转化到了环加外向树上，我们可以在环上切掉一条边，之后用树链剖分维护。

对于环上的所有边，我们都强制设定为重边，这样如果两个点走到了环上，我们就可以特判。

对于询问，在向上走的时候同时维护最大权前缀以及答案即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n)$。

29 MARCH13

29.1 LECOINS

题目大意

有n种类型的硬币，编号从1到n。

每种硬币有颜色C_i和价值V_i。

现在问，如果要恰好组成S美元，最多能有几种不同的颜色？

题解

考虑DP。

首先我们考虑对价值最小的硬币取模的情况，这里设价值最小的硬币价值为V。

之后我们用动态规划处理出数组$f[i][j]$，表示余数为j，用i 种颜色的情况下最少的总价值。

这里的动态规划只要从前往后枚举每一种颜色再用完全背包即可。

转移为:$f[i][j] = min(f[i][j], f[i-1][(j-V_i)\ mod\ V] + V_i)$

注意这里要特判面值最小的硬币颜色的情况。

最后对于一组询问，我们只要二分颜色数量判断总价值是否$\le S$ 即可。

复杂度

时间复杂度$O(Vn + Q\log n)$，空间复杂度$O(Vn)$。

30 FEB13

30.1 QUERY

题目大意

给定一棵n个点的树，有三种操作：

1、树链上加一个等差数列。

2、求树链权值和。

3、回到之前的一个操作。

题解

首先考虑树链剖分。

1、2操作只要使用线段树维护，是经典问题，只要对线段树上每个节点打标记即可。

对于操作3我们只要将树链剖分可持久化，用主席树维护树链剖分即可。

复杂度

时间复杂度$O(n \log^2 n)$，空间复杂度$O(n \log^2 n)$。

31 JAN13

31.1 ANDOOR

题目大意

给定一个矩形边界以及n个圆，求所有圆组成的图形的边界长度。

题解

对于每个圆，我们只要求出他不和其他部分交的边界长度，最后将其加起来即可。

考虑枚举这个圆，之后我们可以将它和其他所有圆以及边界的交，之后按照极角排序，扫一遍即可求出他和其他圆不相交的边界长度。

复杂度

时间复杂度$O(n^2)$，空间复杂度$O(n)$。

31.2 CUCUMBER

题目大意

已知B个$n*n$的矩阵$A_1, A_2, .., A_B$。

对于每对(a, b),$1 \le a < b \le B$构造一个新的矩阵$C_{a,b}$ 满足

$C_{a,b}[i][j] = \sum_{k=1}^{N} A_a[i][k] * A_a[j][k], 1 \le i, j \le N$

对于所有1 n的排列$s = s_1, s_2, ..., s_n$，定义一个新序列:

$F(s)[i] = C_{a,b}[i][s_i], 1 \le i \le n$

如果有奇数个不同的排列s使得$F(s)$中至少有一个奇元素，那么称(a, b)是一个好数对。

统计好数对的个数。

题解

令矩阵$E_{a,b}[i][j] = (C_{a,b}[i][j] + 1)\ mod\ 2$。

之后考虑矩阵$E_{a,b}$的行列式值，发现它在$n > 1$时的奇偶性等价于好数对的要求。

这个时候注意到，如果我们为矩阵A在最后增加完全由1 组成的一行，我们就可以将$D_{a,b}$ 表示成$A_a^T A_a$。

定义矩阵$A_{a,c}$为矩阵A_a删掉第c行得到的矩阵。之后我们应用Cauchy-Binet公式，我们可以得到

$\det E_{a,b}\ mod\ 2 = \sum_{c=1}^{n+1} \det A_{a,c} * \det A_{b,c}\ mod\ 2$

用$n + 1$位二进制数V_a保存所有的$\det A_{a,c}$。

这样$E_{a,b}[i][j]\ mod\ 2$就可以用$V_a \& V_b$中1的个数来表示了。

最后我们可以使用Guass-Jordan消元法快速求出所有$A_{a,c}$ 即可。

复杂度

时间复杂度$O(n^2B + B^2)$，空间复杂度$O(n^2B)$。

32 DEC12

32.1 DIFTRIP

题目大意

给定一棵n个点，以1为根的树，每个点的值为其出度。

统计所有自上向下的不同子串个数。

题解

考虑经典的字符串后缀算法sam（后缀自动机）。

我们只要将sam建立在树上，之后在sam上dp统计即可得到答案。

由于字符集较大，我们可以使用hash来存储每个点的儿子，易证，字符集最大为$O(\sqrt{n})$。

复杂度

时间复杂度$O(n\sqrt{n})$，空间复杂度$O(n\sqrt{n})$。

33 NOV12

33.1 COUNTARI

题目大意

给定n个整数$A_1, A_2, ..., A_n$，求三元组(i, j, k) 的数量，满足$1 <= i < j < k <= n$ 并且$A_j - A_i = A_k - A_j$。

题解

考虑将n个整数分成K块，之后枚举j的值，之后考虑两种情况。

1、i、k有一个和j在同一块中。

这种情况只要枚举那个在同一块中的点的值，再对i的两侧各开一个数组记录每种数出现的次数统计即可。

这部分的复杂度是$O(n * n/K)$。

2、i、k都不在同一个块中。

对于这种情况，一个块内我们可以同一处理。只要将块两侧的所有数通过FFT 统计出每种和的出现次数就可以统计了。

这部分的复杂度是$O(n \log nK)$

只要将K的值定为$\sqrt{n/\log n}$就可以最小化复杂度。

复杂度

时间复杂度$O(n\sqrt{n \log n})$，空间复杂度$O(n)$。

33.2 MARTARTS

题目大意

有两支队伍A队和B队，分别有n个选手参加比赛。

现在给定A队的每一个人和B队的每一个人比赛的得分情况。

A队可以安排n场比赛，每场比赛两方都有一名选手参赛，并且不能有一名选手参加多次比赛。

每次比赛两方由谁参加由A队安排，但是B队可以选择取消一场比赛或不取消。

设A队的得分为H，B队的得分为G，A队的目标是在最大化$H-G$的情况下最大化H，B队的目标是在最大化$G-H$的情况下最大化G。

请给出A队最优的安排比赛方式。

题解

假如B队不取消比赛，我们可以用KM算法求出一组最优解。

由于B队会取消一场比赛，而显然的，B队一定会取消A队比B队得分多的最多的那一场比赛。

所以我们可以把任意两人间的情况作为边从小到大，之后枚举B队取消掉哪一场比赛。我们先将这场比赛的权值设定为无穷大，强制选择这场比赛，进行一次KM 操作求出最大匹配之后更新答案，之后再将权值改回来再进行一次KM操作进行匹配。

之后有几点注意点：

1、对于没有在图中的权值，我们默认它为无穷小。

2、如果需要取消掉的比赛的权值为负，那么B队就不会取消掉它。

3、对于差相同的边权，要将得分较多的排在前面。

这样一次操作的复杂度是$O(n^2)$的，所以满足要求。

复杂度

时间复杂度$O(n^4)$，空间复杂度$O(n^2)$。

34 OCT12

34.1 MAXCIR

题目大意

给出一个三角形ABC，以及N个操作。第i个操作有两个参数x_i, y_i，使用其可使点A的坐标从(X, Y) 变成$(X + x_i, Y + y_i)$。

现在最多可以用K个操作，一个操作不能被重复使用，要求最大化三角形的周长。

题解

由于边BC的长度是确定的，所以只要最大化$|AB| + |AC|$ 即可。

我们一定能找到一对数(u, v)，使得在最大化$uA_x + vA_y$的同时，$|AB| + |AC|$也最大化。假设能达到的最优解为s，它的那组(u, v)一定是椭圆$|AB| + |AC| = s$上那个最优的A点的切线的斜率。

假如已经知道了(u, v)，那么我们将操作排序，二分出前K个操作中为正的操作，之后将其加起来即可。

那么如何知道(u, v)呢？准确的(u, v)并不需要知道，我们只需要将n个操作的所有不同大小关系和正负的情况全都计算一次即可。

对于任意两个操作，我们可以找到两组(u, v)使得他们相等，这样就将平面分成了两个部分，两个向量分别在一侧较大。

对于一个操作，我们也可以找到两组(u, v)使得其为0，这样平面也分成了两个部分，在两侧分别为正和负。

这样我们只要将所有向量全都取出来，之后按照极角排序，每次扫到一个向量时将对应的两个操作的顺序调换同时求解即可。

由于精度要求较高，所以将整数部分和小数部分拆开运算即可。

复杂度

时间复杂度$O(n^2 \log n)$，空间复杂度$O(n^2)$。

34.2 MAXRECT

题目大意

给定一个$n * m$的矩阵，要求找到一个子矩阵使得其权值和最大。

题解

首先我们可以在$O(n)$的时间内求出将一行或一列状态取反对答案的贡献。

之后我们只要用模拟退火算法，每次随机一行或一列，根据贡献随机是否取反。

卡时进行T次即可通过。

复杂度

时间复杂度$O(Tn)$，空间复杂度$O(nm)$。

35 SEPT12

35.1 PARADE

题目大意

给定一张n个点m条边的有向图，每条边有费用。

你可以设置一些人，每个人经过有向图上的一条路径。

最小化费用。费用的计算：

1、沿道路移动的总费用。（如果一条道路被k人次经过，费用就将计算k次）

2、如果一个人的起点和终点不同，需要花费C来送他回家。

3、如果某个点没有被任何一名人经过，花费C。

题解

整理一下题面，发现如下信息：

1、如果一个点有出度，那么他就不会有C的费用。

2、如果一个点有出度，并且它的入度比出度多，这是不优的。因为我们显然可以把某条入度删掉，直接就少了一条边的价值。

综上，可以得到，一个点只有一个入度和一个出度。然后就想到了二分图最小权匹配。

首先先floyd一遍处理出两两点间距离，然后建费用流图。

之后按照询问的C排个序，用spfa跑费用流来算最小权匹配。

每次判断最短路的费用是否小于C，如果是就流，否则就不流。

复杂度

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

35.2 KNGHTMOV

题目大意

有一个骑士，他要从$(0,0)$走到(X,Y)，其中有K个障碍点，不能通过。

骑士有两种走法，可以从(x,y)走到$(x+Ax,y+Ay)$或$(x+Bx,y+By)$。

问有几种不同的方案从起点走到终点。

两种方案不同当且仅当步数不同或两种方案中第i步所在的点不同。

注意：走到终点后可以继续走。

题解

如果(Ax,Ay)和(Bx,By)线性相关，那么我们只要将所有点建出来，跑一边拓扑序之后DP 即可。

如果两者不相关，那么我们考虑容斥。

首先这样我们可以将图建成一张网格图，每个障碍点对应的格点可以解二元一次方程组解得。

考虑$f_{i,j}$表示不经过障碍点走到点(i,j) 的方案数。

由于网格图的规模可能较大，所以我们只对于障碍点处理f 数组。

$f_{i,j} = C_{i+j}^{i} - \sum f_{k,l} * C_{i+j-k-l}^{i-k}$

这样我们只要从左下向右上枚举所有障碍点即可。

答案就是终点的f值。

复杂度

时间复杂度$O(500 * 500)$，空间复杂度$O(500 * 500)$。

35.3 SIMNIM

题目大意

给定n个数，要求将这些数分成几份，要求每一份内的异或值为0。

最大化份数。

题解

考虑从前往后扫，同时用线性基维护当前所有没有被用过的数。

每当一个新的数可以和线性基中的其他数相消得到0的时候，我们就重构线性基即可。

由于时间较为宽裕，所以我们可以多次random shuffle 来得到最优解。

复杂度

时间复杂度$O(n * 64)$，空间复杂度$O(n)$。

36 AUG12

36.1 MAGIC

题目大意

有一张n个点m条边的无向图。

现在有两个魔术师玩游戏。

两个魔术师轮流操作，每个魔术师每次操作时可以选择两个点建一条边。

操作完之后，如果魔术师还有剩余能量，他就可以选择一个节点转移过去。

如果一个魔术师开始操作时，两个魔术师在同一个联通块中，那么这个魔术师就胜利了。

现在给定这张图和初始能量，问先手必胜还是后手必胜。

题解

如果初始两个人就在同一个联通块中那么先手必胜。

如果图中有奇数个节点，那么胜利者直接确定，因为最后肯定是一个奇数块和一个偶数块，连的边数的奇偶性直接确定。

如果图中有偶数个节点，那么按照是否有初始能量分两种情况。这里我们设奇数块的个数为A_1，偶数块的个数为A_2。

在没有初始能量的情况下，分两种情况：

1、先手在两侧都是偶数大小块时胜利，这时只要A_1和A_2 不同为奇就必胜。

2、先手在两侧都是奇数大小块时胜利，这时只要A_1和A_2 不同为偶就必胜。

在有初始能量的情况下，分两种情况：

1、先手在两侧都是偶数大小块时胜利，这时只要A_1不为0 或$A_2 > 2$ 则必胜。

2、先手在两侧都是奇数大小块时胜利，这时在$A_1 \leq 1$ 并且$A_2 = 2$时必胜。

复杂度

时间复杂度$O(n+m)$，空间复杂度$O(n+m)$。

36.2 GTHRONES

题目大意

有n个数字，两个人轮流操作。

在第一轮中，先手从可选数中选择一个数作为X，并将其删除，之后两人轮流操作。

每一轮操作者要从剩余可选数中选择一个数，这个数和X 只相差一个质因子，之后用这个数替代X，并将其删除。

现在问先手必胜还是后手必胜。若先手必胜，输出第一轮中可选的最小的X。

题解

根据质因数个数，我们可以将这n个数分成两份成为一张二分图。

如果这张二分图有完备匹配，那么后手必胜。

因为不论先手选哪个数，后手一定可以选择和这个数匹配的那个数。

如果没有完备匹配，那么对于一个点，如果有一个完备匹配不包含这个点，那么这个点一定是先手必胜点，因为这个时候不论后手选哪个点先手都能找到一个匹配点。

用匈牙利算法即可解决问题。

复杂度

时间复杂度$O(nm)$，空间复杂度$O(n+m)$。

37 JULY12

37.1 DGCD

题目大意

给定一棵有n个节点的树，每个节点有权值，要求支持两种操作。

1、求树链上所有点权值的gcd。

2、将树链上所有点的权值加上一个值k。

题解

考虑树链剖分。

对于每一个点，如果他在他父亲的重边上，那它的权值就是它父亲的权值减去他，否则就是它本身的权值。

操作1：只要在线段树上维护区间的gcd值，之后将对应树链的所有结点值gcd 一下就可以得到答案。这里注意，lca 点必须使用原有权值参与gcd。

操作2：只要在每次走轻边时修改两个点的权值即可。

复杂度

时间复杂度$O(n\log^2 n)$，空间复杂度$O(n)$。

37.2 EST

题目大意

给定一个长度为n的字符串，问有多少个不同的字符串所构成的后缀字母树与该字符串的后缀字母树同构。

题解

首先，对于一棵后缀字母树，我们一定能找到其中最长的，没有在后缀树中出现的一条链。这部分可以使用后缀数组做到。

易证，所有比这条链短的链都不会再后缀字母树中出现。

如果找到了这条链所在的位置，那么所有比它短的链的位置都确定了。

所以问题转化为了找到这条链所在的位置。

考虑一个确定后缀，如果它删去和其他确定后缀的公共前缀之后，比最长不确定后缀要短，那么最长不确定后缀较长的那部分就确定了。这部分可以直接利用$height$数组解决。

最后，我们可以将最长不确定后缀可以出现的位置用后缀数组上的一段区间表示。之后将所有$height <$最长不确定后缀长度的后缀都统计入答案。

最后按照不同的字母数量乘上置换数即可。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

38 JUNE12

38.1 MATCH

题目大意

有一张左侧n个点，右侧m个点的二分图。

对于左侧第i个点和右侧第j个点，有$f_{i,j}$的几率有一条边。

求期望二分图最大匹配数。

题解

对于左侧的n个点，我们可以用一个n位的二进制数表示匹配情况，每一位表示对应点是否被匹配。

之后我们可以用一个2^n位的二进制数表示每一种匹配情况是否可能，这样就可以完整地描述左侧节点的匹配情况了。

我们可以先通过dfs预处理出所有可能的2^n位二进制数以及之间的转移，之后使用DP算出答案。

通过打表可得，在$n = 5$时，最多只有406种不同的情况。

复杂度

时间复杂度$O(mP)$，空间复杂度$O(mP)$，其中P为可能的2^n 位二进制数的个数。

38.2 COOLNUM

题目大意

一个数A有k位，从中选出至多3个不同的数位，令这几位的和为S，并且这个数的所有数位和为K。

如果存在一种选取方案满足$(K-S)^S$是A的倍数，那么就把A称为cool number。

现在给定一个数n，求小于等于n的最大的cool number和大于n的最小的cool number。

题解

cool number分为两类，第一类只有不超过3个非0位，这一类可以很方便得求得，所以我们只考虑剩下的cool number。

由于$S \leq 27$，所以我们可以得到$(9k-27)^27 > 10^{k-1}$，解出k的上界为77，所以我们只要枚举$(K-S)$，将其分解质因数，求出$(K-S)^27$所有满足条件的因数，再判断是否和K, S吻合。

通过打表我们发现$K-S$的上界只有350，满足条件的数的个数只有30000多个，所以直接暴力找出所有数再处理询问即可。

注意重复的数是不需要去重的。

复杂度

时间复杂度$O(350*27*P)$，空间复杂度$O(S*77)$，其中P为1到350中最多的质因子数，S是满足条件的数的个数。

38.3 CLOSEST

题目大意

三维空间中有n个点。现在有q组询问，每组询问给出一个点的坐标，询问在n个点中距离这个点最近的点的编号。

题解

这题可以用kd-tree达到百分百的准确率，稍微卡常之后就可以通过了。

需要注意的是要用unsigned long long来存距离的平方。

复杂度

时间复杂度$O(n^{\frac{5}{3}})$，空间复杂度$O(n)$。

39 MAY12

39.1 LEBOXES

题目大意

有n个盒子，m个物品。

第i个盒子有P_i的概率有V_i块钱，剩下$1 - P_i$的概率有1 颗钻石。

第i个物品需要花费C_i块钱和D_i个钻石。

问期望能买几个物品。

题解

首先使用DP预处理，$f[i][j]$表示i个钻石，j 个物品最少要花多少钱。

之后考虑将物品分成两份分别dfs出所有可能的钻石物品数量组合以及他们的概率。

最后枚举两边使用的钻石数以及最终的物品数，之后将两份合并起来即可。

复杂度

时间复杂度$O(2^{n/2} * n^2 * m)$，空间复杂度$O(2^{n/2})$。

39.2 TICKETS

题目大意

给定一张图，求一个最小的点集，满足点集内边数大于点数。

题解

考虑满足题意的几种构图方式。

1、两个点之间有三条路径。

2、两个环用一条链连在一起。

对于第一种情况，我们只要枚举这两个点，之后用bfs 算出两点之间的三条最短路，求出他们的和。

这里只要保证最后一条边不走重即可。如果在中途路径有相交，那么保证可以找到一组更优的解。

对于第二种情况。我们可以枚举一个链上点，以他为根建出一棵bfs 树。这个时候每一条非树边就对应一个环。

规定一个环的权值为它的大小加上他和根的距离，这时我们只要求出权值最小的两个环，再将他们加起来就是一组解。

这里不用担心路径的相交，因为如果相交，那么就一定有更优的解。

复杂度

时间复杂度$O(n^2 * m)$，空间复杂度$O(n + m)$。

40 APRIL12

40.1 CONNECT

题目大意

给定一张$n * m$的地图每一格有两个权值$a_{i,j}$和$b_{i,j}$，要求找一个联通块包含

至少k 种非零$a_{i,j}$ 权值，使得联通块中$\sum b_{i,j}$最小。

联通块中不能包含$a_{i,j} = -1$的点。

输出最小值。

题解

由于数据范围较大，考虑随机化。

我们可以将每个$a_{i,j}$权值对映到$[1, k]$ 中，之后使用状压DP 来解决问题。

这样显然不会让答案变大。

由于有$k!/k^k$的期望得到答案，所以我们只要进行$k^k/k!$ 次随机，基本就能得到答案了。

复杂度

时间复杂度$O(k^k/k! * 3^k * n * m)$，空间复杂度$O(2^k * n * m)$。

40.2 TSUBSTR

题目大意

给定一棵树，要求求这棵树上所有自上而下的子串的个数，以及在每个字母在给定大小顺序下的第K大的子串。

题解

考虑使用后缀自动机解决问题。

由于是自上而下的子串，所以我们只要把每个点接到他的父亲上即可。

之后使用简单的DP即可求出所有不同子串个数。

对于第二问，由于保证输出量的大小，所以我们只要用DP 求出每个点开始能接受的不同子串个数之后在sam上走即可。

复杂度

时间复杂度$O(26 * n + \sum output)$，空间复杂度$O(26 * n)$。

40.3 SIMGRAPH

题目大意

给定两张n个点的无向图，要求将第二张图的点重新编号，使得两张图的相似度最高。

相似度的定义为同时存在在两张图的边数。

题解

首先我们可以在$O(n)$的时间内求出交换两个点编号后对答案的贡献。

之后我们使用模拟退火算法进行T次，每次随机两个点的编号计算贡献后根据贡献随机是否交换即可。

复杂度

时间复杂度$O(n*T)$，空间复杂度$O(n^2)$。

41 MARCH12

41.1 EVILBOOK

题目大意

有n个人，每个人有两个值A_i, B_i。

你可以付出A_i的代价打败一个人获得B_i的能力值。

你可以支付K的能力值来使一个人的两个值都变成原来的1/3。

一个人只能被打败一次，并且随时保证你的能力值非负。

求将你的能力值提升到大于等于666最少需要多少代价。

题解

考虑搜索。

1、如果打败一个人得不偿失，我们一定不会去打他。

2、我们肯定先打那些不需要支付能力值的人。

3、如果剩下的所有人的能力值加上当且能力值的和已经不能大于等于666，直接return。

4、如果当前的代价已经超过搜到的最小答案，直接break。

综上，我们可以按照需要削弱的次数来进行搜索。

先搜索不削弱的，之后搜索削弱一层的，以此类推。

然后在搜索时，进行一些剪枝即可。

复杂度

时间复杂度$O(n^3 * 666)$，空间复杂度$O(n)$。

41.2 CIELQUAK

题目大意

给定一张$R * C$的网格图。任意两个相邻点之间都有一条边。

现在每条边都有p的概率被损坏，问有多少的概率可以从$(1,1)$ 点走到(R,C) 点。

已知p不小于0.1。

题解

在C比较小的时候，我们可以使用轮廓线DP来求解。

首先我们可以用dfs预处理出所有可能的轮廓线以及转移，发现只有3000多种。

之后使用DP计算即可。

在C比较大的时候，我们发现答案满足关系$ans(R,C,p)/ans(R,C+1,p) \approx ans(R,C-1,p)/ans(R,C,p)$。

所以我们可以只计算到C=50的时候，后面的部分用快速幂近似求解即可。

复杂度

时间复杂度$O(TRxS)$，空间复杂度$O(RxS)$ 其中S为不同的轮廓线方案数。

42 FEB12

42.1 FINDSEQ

题目大意

给定一个长度为n的数组$A_0, A_1, ..., A_{N-1}$，要求找出一个长度为5 的子序列，其中元素的大小顺序满足给定字符串B的5 个元素的大小顺序。

题解

考虑枚举第二个和第四个元素出现的位置。

之后我们从大到小来确定剩下的3个元素，我们可以确定这三个元素的上界。由于这三个元素的区间已经确定且互不相交，所以取最大可行解一定是最优的。

所以我们可以用数组维护前缀和，用二分来确定区间内比某个值小的最大值。

之后再扫一遍5个元素判断是否满足条件即可。

复杂度

时间复杂度$O(n^2 \log n)$，空间复杂度$O(n^2)$。

42.2 ROC

题目大意

给定一个$n * m$的网格图，描述了一个房间。

在每个90度内角处有一个小朋友。每个小朋友只能和相邻的小朋友交换位置。

在移动时，小朋友贴着墙移动。小朋友移动一格需要1的时间。

一个小朋友只能同时与至多一个小朋友交换位置。

现在给定两个小朋友，问他们要相遇至少要多少时间。

题解

我们从上向下枚举每一行，分别记录从左上角走到每一行最左侧和最右侧的步数。

借此我们可以算出每两个相邻的小朋友之间的距离以及小朋友的顺序。

之后对于一组询问，我们可以确定的是，这两个小朋友一定是先移动到两个相邻的位置，之后再交换位置。

通过计算发现，这两个相邻的位置一定是中点最接近两个小朋友距离中点的那两个位置。

所以通过二分即可确定位置，之后计算答案即可。

复杂度

时间复杂度$O(n*m+n\log n)$，空间复杂度$O(n*m)$。

42.3 FLYDIST

题目大意

有n个城市，m条直达航线，每条航线需要一定的时间z_i。

现在要求你给第i航线增加d_i，或减少e_i的时间，使得没有两条路满足一条比另一条需要坐更多次飞机却时间更短。

要求最小化$\sum d_i+e_i$

题解

我们考虑单纯形算法。

用变量$b_{i,j}$表示从点i走到点j最短的时间，那么就有约束$b_{i,j}+b_{j,k}\geq b_{i,k}$，对于所有的直达航线$x_i,y_i$，都有$b_{x_i,y_i}=z_i+d_i-e_i$

但是由于这样初始解不满足条件，所以我们必须对其进行一些变形。

我们可以将所有边的初始时间设定为20，这样我们就需要求出点i走到点j的最小航班数$c_{i,j}$，然后我们设$b'_{i,j}=20*c_{i,j}-b_{i,j}$，用$20*c_{i,j}-b'_{i,j}$取代$b_{i,j}$，再用$19-d_i$和$z_i-e_i-1$替换$d_i$,$e_i$，这样所有的初始约束就满足了，就可以利用单纯形算法求解了。

最后求出的答案是一个小数，所以我们要枚举分母将其化为分数。

复杂度

时间复杂度$O(单纯形(n^2, n^3))$，空间复杂度$O(n^5)$。

43 JAN12

43.1 CARDSHUF

题目大意

定义操作(a, b, c)为以下6步

1、先将数列前a个数取出。

2、取出数列前b个数。

3、将操作1的a个数放回数列前。

4、将数列前c个数取出。

5、将操作2的b个数左右翻转之后放回数列前。

6、将操作4的c个数放回数列前。

初始给定长度为n的数列A满足$A_i = i$。

之后给定m个操作，求所有操作结束后的数列A。

题解

由于题目中有翻转、合并、分离操作，所以我们可以使用平衡树来解决这道题。

只要用平衡树按照题目的要求模拟即可。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n \log n)$。

43.2 MISINT2

题目大意

如果一个长度为n的只包含小写字母的字符串s，如果把它的偶数位按顺序放前面，奇数位按顺序放后面，依然和之前一样，那么就称它是好的。

问有多少个长度在$[L, R]$之间的字符串是好的。

题解

发现重排操作就是一个置换，对于长度为n的字符串，我们只要统计出置换中的环的个数g_n，答案就是26^{g_n}。

那么问题就变成了求环的个数g_n。

定义$ord(p)$为2模p的阶，我们发现$g_n = \sum_{p|(n+1),p>1} \frac{\phi(p)}{ord(p)}$。

首先我们可以计算出1到$\sqrt{R}$之间的所有质数，之后我们枚举每个质数，将$[L, R]$中所有该质数倍数的数都筛出来，这样就得到了每个数的质因数分解。

之后我们对于所有$\le \sqrt{R}$的数预处理出其ϕ 值，再对$\le \sqrt{R}$的质数的幂求出其ord值。

由于在a, b互质时，有$ord(ab) = lcm(ord(a), ord(b))$，所以我们可以在dfs数$n$的因数的同时求出$\phi$值和$ord$值。

对于每一个数，至多有一个质因子P的ord值没有被求，我们只需要枚举$P-1$的所有质因子即可得到答案。

最后按照式子统计答案即可。

复杂度

时间复杂度$O(\sqrt{R}\log R + (R-L)(\frac{\sqrt{R}}{\log R} + \log^2 R) + \frac{R^{\frac{3}{4}}}{\log R})$，空间复杂度$O(\sqrt{R}\log R + (R-L)\log R)$。

44 DEC11

44.1 SHORT2

题目大意

给定一个质数p，求数对(a,b)的个数要求满足$a*b$ 被$(a-p)*(b-p)$整除。

题解

题目等价于求数对(a,b)的个数，满足$a*b|p*(a+b+p)$。这时可以分三类情况讨论。

1、a,b同时被p整除。对于这种情况只有五种不同的数对$(a/p,b/p)$:

$(1,1),(1,2),(2,1),(2,3),(3,2)$。

2、a,b只有一个被p整除。对于这种情况，我们可以将数对(a,b) 变形为$(a,p*(p+a)/b)(p|b)$或$(p*(p+b)/a,b)(p|a)$。变形之后就是3的情况，所以3的一种情况对应2的两种情况。

3、a,b都不被p整除。对于这种情况，在$a=b$时，只有$(1,1)$ 一组解。在$a\neq b$ 时，我们可以假设$a<b$。由于$a*b\leq(a+b+p)$,所以我们可以得到$a<sqrt(p+1)+1$。在这个条件下，我们可以得到$b=\frac{a+p}{ka-1}$。我们设$d=ka-1$，那么一个合法的解必须满足：$1.b>a2.a$不被p 整除$3.d|(a+p)4.a|(d+1)$ 由于d和b的上界都是$\sqrt{p}$级别的，所以我们只要枚举b 和d，即可求得答案。

复杂度

时间复杂度$O(T\sqrt{p})$，空间复杂度$O(1)$。

44.2 HYPER

题目大意

一个3−超图类似一个普通的图，只是其中的边都连接三个点。

一个3−超树是删去任意一条边都不连通的3−超图。

给定N，问有几种不同的3−超树。

题解

由于数据范围较小，所以可以打表。

对于一棵3−超树，我们可以把它分成若干个点双连通分量。

之后我们发现对于一个点双联通分量，每一条边连接的三个点中有且仅有一个仅属于这条边。如果我们将这些点删去，就得到了一张正常的点双联通图。原来3− 超图的点数对应这个图的点数+边数。

我们可以暴力搜索点数为i的双联通超树数量，接着暴力枚举每一个点双联通分量的大小，之后用记忆化搜索拼接即可。

最后在本地将数据跑出来交表即可。

复杂度

时间复杂度$O(T)$，空间复杂度$O(n)$。

45 NOV11

45.1 STEPAVG

题目大意

现在给定n个数，每次可以从中拿出两个数，求出他们的平均数后再放回。

进行$n-1$次操作之后，就只剩下一个数，你要使得这个数尽量接近K。

题解

首先将和K差较小的数取出，之后将剩下的m个数排序，设排序后第i个数的值为a_i。

定义函数$get(l,r,k)$表示使得$[l,r]$区间中的数操作到最接近k。

这里我们先比较一下第l个数和第r个数哪个距离k 较近。

假如第l个数距离k较近，我们就调用$get(l+1,r,k*2-a_l)$，反之调用$get(l,r-1,k*2-a_r)$。

这样我们只要调用$get(1,m,k)$,之后再和所有取出的数合并即可。

复杂度

时间复杂度$O(n)$，空间复杂度$O(n)$。

45.2 LUCKYDAY

题目大意

有一个数列S：

$S_1 = A$

$S_2 = B$

对于$i \geq 3$有$S_i = (X * S_{i-1} + Y * S_{i-2} + Z)\ mod\ P$

现在给定A, B, X, Y, P, C，之后有Q组询问，每组询问包含L_i和R_i，表示问i在L_i到R_i之间，有多少$S_i = C$。

题解

首先特判$Y = 0$的情况，这部分只要暴力找出循环节即可。

之后考虑如何找出循环节，显然这里的循环节长度是小于等于P^2的。

我们可以用BSGS算法，考虑将$(S_{j-1}, S_j)(j \leq K)$全都塞入hash表中。之后枚举iK，询问hash表中是否有(S_{iK-1}, S_{iK})，如果有，说明找到了循环节，长度为$iK - j$。

之后我们清空HASH表，并将$(S_{iK-1}, S_{iK}(i \leq \frac{P^2}{K})$塞入HASH表中。

之后我们枚举$0 \leq i \leq P$，并求出(C, i)以及其后面的K项，并在HASH表中查询。若查出结果，那么就将其出现的位置记录下来。

定义函数$get(X)$表示求$\leq X$的答案。这个可以利用循环节和每个答案出现的位置简单计算。

对于一组询问，只要输出$get(R) - get(L-1)$即可。

这里K取到$\sqrt{P}$时，复杂度最低。

复杂度

时间复杂度$O(P\sqrt{P} + Q\log P)$，空间复杂度$O(P\sqrt{P})$。

46 OCT11

46.1 PARSIN

题目大意

定义函数$f(M,N,X)$为：

$f(M,N,X)=\sum_{k_1+k_2+,,+k_M=N} sin(k_1X)sin(k_2X),,sin(k_MX)$

其中$k_1,k_2,,,,k_M$都是非负整数。

现在给定M,N,X，请你求出$f(M,N,X)$。

题解

首先，我们可以得到递推式：

$f(M,N,X)=\sum_{i=0}^{N} sin(iX)*f(M-1,N-i,X)$

通过和角公式，我们可以得到递推式：

$f(M,N,X)=f(M-1,N-1,X)sin(X)+2*f(M,N-1,X)cos(X)-f(M,N-2,X)$

之后使用矩阵乘法加速计算即可。

复杂度

时间复杂度$O(M^3\log N)$，空间复杂度$O(M^2)$。

46.2 BAKE

题目大意

有n个操作，每一个操作都可能是增加订单或者询问，每一个订单用如下的方式描述：

I 产品编号[大小编号] 省编号[城市编号[地区编号]] 性别 年龄 出售数

顾客的性别为M或者F，年龄从1到90。注意所有的编号都是从0开始。还需要注意方括号内的部分是可选的，因为某些出售的这些信息丢失了。出售数量不会超过100。

每一个询问都可以用如下的方式描述：

Q 产品编号[.大小编号] 省编号[.城市编号[.地区编号]] 性别 起始年龄[-结束年龄]

询问该限制下的出售总数。如果方括号内的信息缺失说明对这个信息没有限制，特殊地如果产品编号省编号是-1 说明没有限制。

题解

开一个7维数组来记录所有的订单，用0来表示没有限制。

对于年龄一维，用树状数组维护前缀和即可。

复杂度

时间复杂度$O(1)$，空间复杂度$O(1)$。

47 SEPT11

47.1 SHORT

题目大意

给定n, k，求满足$n < a, b < k$的数对(a, b)使得$(a-n)(b-n)|ab-n$。

题解

假设$a \geq b$，那么可以得到$b = n + \frac{n(a-n)}{p(a-n)-a}$。

之后枚举所有可能的a，再枚举$n(a-n)$的所有约束d，这样就可以求得$p = \frac{d+a}{a-n}$和b，判断是否合法之后就可以累加进答案了。

由于$d + a \geq 2(a-n)$，所以就有$b \leq n + \frac{n(a-1)}{p(a-n)-a}$，由因为$a \leq b$，所以可以得到$a \leq 2n + \sqrt{2n^2 - n}$。于是$a$就只需要枚举$O(n)$个值即可。

对于枚举d，我们可以预处理出所有可能的$a-n$的质因数分解，然后将$n(a-n)$的所有约束求出并进行判断。然而这样复杂度还是较高。

当a较大的时候，由于对于$a \leq b$可以得到$p \leq \frac{a^2-n}{(a-n)^2}$，所以可以直接枚举所有可能的$p$，然后解出$d$判断是否合法。

复杂度

时间复杂度$O(TnS)$，空间复杂度$O(n\log n)$，其中S 上界为1200。

47.2 CNTHEX

题目大意

给定四个数n,l,x和k。

现在求满足最长边长度$\in [l,n]$，除了最长边每条边长度都不超过x，并且没有超过k条相同长度的边的六边形个数。

题解

由于$n-l$较小，所以我们可以考虑枚举最长边的长度y，之后统计剩下边的长度。

考虑数位DP。

首先我们将5条边按照非升顺序排列。对于每一条边，都有一条大于等于它的边，最大的哪条边小于等于x。

从高向低枚举每一个数位，我们用$f_{i,j,k}$，i表示第i个数位，j表示y的前i位与5个数的前i 位的差，二进制数k表示5个数中每个数和前一个数的大小关系。

由于如果j大于等于5就不可能有结果，所以j必须满足小于等于4。

对于转移，我们可以新建一个数组$g_{j,k,l}$，其中j 和k与之前f中相同，l则表示上一个数这一位选了啥。

之后我们从大到小枚举这5个数，之后枚举这一位转移即可。

最后将$f_{i,j,k}$中合法的部分取出累计入答案即可。

复杂度

时间复杂度$O(\log x)$，空间复杂度$O(\log x)$，常数巨大。

48 AUG11

48.1 SHORTCIR

题目大意

对于一个布尔型表达式，一些值的确定可以让我们不用计算一些其他值，这样可以大大节省计算时间。

现在给定一个长度为n的布尔型表达式以及其中每个变量为1 的概率，现在要求你制定计算顺序，问期望最小计算期望是多少。

题解

首先我们将表达式建成一棵树，之后从下向上DP。

对于一个节点，我们记录两个信息:为1的概率A_i和期望计算次数B_i。

如果一个节点的儿子都用and连接起来，那么我们按照$B_i/(1-A_i)$ 从小到大排序，之后从前往后计算。

对于一个节点的儿子都用or连接起来，那么我们按照B_i/A_i 从小到大排序，之后从前往后计算。

这里注意，如果一个节点前有not，需要执行$A_i = 1 - A_i$。

最后将根节点的B_i输出即可。

复杂度

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

48.2 DIVISORS

题目大意

对于给定的正整数B和X，求满足条件的正整数N的个数：要求对于N，至少存在一个数$D(N < D \leq B)$能整除$N * X$。

题解

令$S = \frac{N*X}{D}$，显然得，我们可以得到$S < X$。

由于一个N会被不同的S同时计算，所以我们必须要去重，这里我们可以只在最大的S处统计。

因为$S|N*X$，所以$\frac{S}{Gcd(S,X)}|N$，令$F_S = \frac{S}{Gcd(S,X)}$，由此令$L_S = \frac{B*S}{X*F_S}$我们可以得到$N \le L_S$。

假设N同时会在S,T两处被计算并且有$S < T$，那么有$T|N*X$，令$G_T = \frac{T}{gcd(F_S,X,T)}$所以我们得到$G_T|\frac{N}{F_S}$。

所以我们可以枚举S，之后对于所有的T进行容斥，将多余状态删去。

直接计算会超时，所以我们可以用hash表将相同的状态都合并起来，之后再将X相同的数据合并计算即可通过。

复杂度

时间复杂度$O(X^2K)$，空间复杂度$O(XK)$，其中K 是不同lcm的数量。

49 JULY11

49.1 YALOP

题目大意

给定一个由$n*m$个格子构成的房间，每个格子有两种权值——0 或1。

每当你离开一个格子时，就会将这个格子以及和他直接相连的四个格子的权值异或1。

现在你在$(1,1)$，你要走到(n,m)。求是否可以在走完之后，保证所有格子的权值都是0。

$1 <= n,m <= 10^9$

$min(n,m) < 40$

题解

首先考虑n、m都不等于1的情况。

对于一个权值为1的点，我们可以通过变换将它转移到最左边一列上。

通过寻找规律，我们发现在这个点转换出的最左边一列根据他的x值是一个大约几千大小的循环。

我们只要暴力找出循环，之后再将所有的权值为1的点转移到最左侧。

之后考虑如何消除最左侧的格子。由于只有在最右边一列进行修改才能全部转移到最左边一列，所以我们就对于最右边一列每种修改求出他对最左侧的影响，之后用高斯消元判断是否可以消完即可。

易证，在n、m均不为1的情况下，只要能消完一定有行走方案。

之后考虑$n = 1$或$m = 1$的情况。

在这种情况下，一次修改相当于修改连续的三个权值。

那么对于一个权值为1的点，我们就可以通过两次修改将它转移到和他距离3个位置的格子。

所以我们只要将所有权值为1的点对应到$[1, 3]$区间内再判断是否可以消完。

由于在这种情况下，消完不一定有解，所以我们要根据操作的奇偶性判断是否有解。

复杂度

时间复杂度$O(min(n, m) * 100000)$，空间复杂度$O(min(n, m) * 100000)$。

49.2 BB

题目大意

有一个长度为n的01序列，它的任意长度为m的连续子序列中都有k个1。蚊所有满足条件的01序列中，1的数目最少的不同的序列有多少个？

题解

在m整除n的情况下，我们可以将串分成$\frac{n}{m}$段，每一段中都有k个1。

我们可以用一个矩阵来表示这个串。第i行第j列表示第i段的第j个1在这段中的位置。这样我们得到了一个半标准杨氏矩阵。

利用杨氏矩阵的计算公式，我们发现$ans = \frac{\prod_{i=1}^{k}\prod_{j=1}^{\frac{n}{m}} i+j+k-1}{\prod_{i=1}^{k}\prod_{j=1}^{\frac{n}{m}} i+j-1}$

直接计算会超时，所以我们可以将上下相同的部分消去，之后就只剩下k^2个不同的数了。

当m不整除n的时候，分两种情况讨论:

1、$n \ mod \ m \le m - k$，那么每一行中都不能选小于$n \ mod \ m$的数，也就是缩小了可选数的个数。

2、$n \ mod \ m > m - k$，那么矩阵中每一行最后$m - n \ mod \ m$个数都被确定了，也就是缩小了矩阵的大小。

复杂度

时间复杂度$O(mk)$，空间复杂度$O(1)$。

50 JUNE11

50.1 CLONES

题目大意

我们称一个形如$f : A \to B$的函数叫做布尔函数，其中A 是一个长度为n的仅由0和1组成的数列的集合，B是由0和1组成的集合。我们称n 为布尔函数的项数。现在有四个是n项布尔函数的集合：

1、Z集合是所有满足$f(0, 0, ..., 0) = 0$的集合。

2、P集合是所有满足$f(1, 1, ..., 1) = 1$的集合。

3、D是满足$!f(x_1, x_2, ..., x_n) = f(!x_1, !x_2, ..., !x_n)$的集合。

4、A集合是满足如下条件的集合：

如果$f(x_1, ..., x_{i-1}, a, x_{i+1}, .., x_n) = f(x_1, ..., x_{i-1}, b, x_{i+1}, .., x_n)$

那么$f(y_1, ..., y_{i-1}, a, y_{i+1}, .., y_n) = f(y_1, ..., y_{i-1}, b, y_{i+1}, .., y_n)$

对任意y成立。

现在给出一个表达式，这个表达式的结果是一个以函数为元素的集合，求出这个集合的元素个数。

题解

首先我们可以用一个4位二进制数来表示一个函数是否属于每一位所代表的集合。

通过打表找规律，我们可以求出这16种不同的n位函数分别有多少个。

之后我们可以用一个16位的二进制数来表示每一种函数是否在集合中。

这样我们就把集合运算转化成了二进制运算，模拟一下表达式计算即可求出整个表达式的情况，之后统计答案即可。

复杂度

时间复杂度$O(T|S|)$，空间复杂度$O(T|S|)$。

50.2 MINESREV

题目大意

现在给定一张已经被打开的R*C的扫雷的地图，每次操作你可以选择一个点x关闭它，同时关闭所有可能同时与x打开的方块。

最小化操作数打开所有的方块。

题解

首先我们将所有没有数字的方块以及其周围一圈有数字的方块称为一个部分。

对于一个有数字的方块，它最多同时被两个部分包含，而我们关闭它就能同时关闭这两个部分。

所以我们处理出所有块之间的同时关闭关系，之后用带花树算法求出这张普通图的最大匹配，用总块数减去之就能求出关闭所有部分的最少关闭次数。

之后将所有不属于任何部分的方块一个一个关闭即可。

复杂度

时间复杂度$O(R^2C^2)$，空间复杂度$O(RC)$。