IOI2016中国国家集训队第一次作业
试题泛做

宁波市镇海中学 邹逍遥

Contents

1 传统题2015

1.1 AUG15 DISTNUM

Description

给定一个长度为n的数组和m个操作，每个操作可以：

- 令S为由下标范围从l到r的不同的元素构成的集合,求$\Sigma_{1\le i<j<k\le |S|} S_i S_j S_k \bmod (10^9+7)$
- 修改一个位置的数的值
- 删除一个数
- 插入一个数
- 询问一个区间内出现过的数字种数

$m, n \le 10^5$

Analysis

首先可以看出第一个询问只需要查询出满足条件的数的$0\sim 3$次方的和就可以计算了，而询问5就是询问0次方和，所以可以忽略询问5。

首先模拟一遍操作，使用平衡树维护所有数的顺序，那么就可以给每个数分配一个不超过m的下标了。

分配下标以后就可以用树状数组套线段树来维护了，查询时就是查询那些l到r之间并且下一个相同的数在r之后的数的和。当有数变化时使用$m+n$个set维护所有相同值的数的集合就可以做到$O(\log n)$查询后一个同值的数。

时间复杂度：$O(m\log^2 n)$

空间复杂度：$O(m\log^2 n)$

1.2 JULY15 EASYEX

Description

将一个k面的骰子投掷n次，设第i面总共出现了a_i次，求$\prod_{i=1}^{l} a_i^f$的期望。输出答案模2003的结果。

$n, k \leq 10^9, l \leq k, l * f \leq 50000, f \leq 1000$。

Analysis

设$x_{i,j}$为第j次是否投到i，那么答案就是$\prod_{i=1}^{l}(\Sigma_{j=1}^{n} x_{i,j})^f$。将这个式子展开以后每一项可以分开计算。

首先消去对于同一个j出现了多个i乘起来的项，那么剩下的项的贡献就是$\frac{1}{k^t}$，其中t为出现的不同变量数。那么只需要预处理出每个t之前的系数就可以计算了。

设$w_{i,j}$为$f = i, t = j$时的系数，那么有递推式：$w_{i,j} = w_{i-1,j-1} + w_{i-1,j} * j$。计算$w$的复杂度是$O(f^2)$。

令$dp_{i,j}$为考虑前i个数字和前j个位置时的答案，那么有$dp_{i,j} = dp_{i-1,j-k} \times f_k \times \binom{j}{k}$。用FFT和快速幂优化即可做到$O(lf \log l \log f)$。

由于n超过模数的那部分对答案的贡献显然是0，所以将n缩小后，可以不使用FFT直接通过本题。

时间复杂度：$O(lf \log l \log f \log n + f^2)$

空间复杂度：$O(lf)$

1.3 JULY15 HAMILG

Description

两个人在一张n个点的图上玩游戏，一开始棋子在其中一个点上，然后两个人轮流移动棋子，谁先经过之前走过的点就输了。求可以使后手获胜的起始点个数。

$n \leq 2000, m \leq 3 * 10^5$

Analysis

可以证明一个点是后手胜利点当且仅当删掉这个点以后原图的最大匹配不变。暴力枚举的复杂度是$O(mn^2)$的。

那么首先使用带花树求出这张图的一个最大匹配，然后对于每个非匹配点都找一次增广路，将所有访问到的点设为可行即可。

时间复杂度：$O(mn)$

空间复杂度：$O(mn)$

1.4 JUNE15 CHEFBOOK

Description

给出m个限制，每个限制给出5个参数x, y, w, l, r，要求确定n个非负整数X_i和n个非负整数Y_i，要求对于每个限制都满足$l \le X_x - Y_y + w \le r$。同时你需要最小化$\Sigma X_x - Y_y + w$，输出一种可行方案或无解。

$n \le 100, m \le 10000$

Analysis

把每一个限制拆开，就得到了$2m$个形如$X_i - Y_j \le k$的限制，可以转化为标准型线性规划。注意到它的矩阵表示中每行仅有一个1和一个-1，可以通过将该线性规划转置，就可以建出$O(n)$个点$O(m)$条边的费用流模型，可以解出答案。

使用费用流解出答案以后，可以看出对于每条有流量的边，对应的约束都达到限制。那么建立一个差分约束系统跑一遍最短路构造一个解即可。

时间复杂度：$O(nm^2)$

空间复杂度：$O(m)$

1.5 MAY15 CBAL

Description

定义“平衡串”是一个长度为偶数且所有字符都出现偶数次的串。

给定一个小写字母组成的长度为n的字符串A，给出m个询问，每个询问给出$x, y, type$，要求$\sum_{x \le l < r \le y, A_{l,r} \text{ is balanced}} |A_{l,r}|^{type}$。

$n, m \le 10^5, type \le 2$。

Analysis

将每个字母表示成2^i，那么一个串是平衡的即这些字母异或起来为0。所以记下前缀异或和之后，需要统计的就是$\sum_{x \le l < r \le y, B_l = B_r} (r-l)^{type}$。

将序列分块，那么可以$O(n\sqrt{n})$的时间预处理出任意两个块端点之间的答案。在询问的时候只需要统计块外部的答案即可。

记下前缀和（需要0次，1次，2次都记）以后一个外部点和一个内部点产生的贡献可以直接计算，而两个外部点产生的贡献只需要像预处理一样一起扫一遍即可。

时间复杂度：$O(n\sqrt{n})$

空间复杂度：$O(n\sqrt{n})$

1.6 MAY15 GRAPHCNT

Description

给定一张n个点的有向图，求有多少对点(x, y)满足存在两条从1到x和1到y的路径不相交。

$n \le 10^5$。

Analysis

首先以1为根进行DFS，并删除那些无法到达的点。那么剩下的图就成了一张以1为根的Flow Graph。

计算出以1为根的Dominator Tree之后，要求的点对其实就是满足$lca(x,y)=1$的点对，直接在Dominator Tree上DP即可。

时间复杂度：$O(n\alpha(n))$

空间复杂度：$O(n)$

1.7 APRIL15 BWGAME

Description

给定一个$n\times n$的矩阵V，第i行的L_i到R_i之间的格子是黑色的，剩下的格子是白色的。

两个人轮流行动，每次每个人需要给出一个全排列A_i，满足$V[i][A_i]$都是黑色的。其中第一个人还需要保证全排列的逆序对数为偶数，第二个热保证为奇数。给出的全排列不能重复。谁先无法给出谁就输了，同时无法给出就平局。求第一个人胜利还是第二个人胜利还是平局。

$n\le 10^5$。

Analysis

容易发现只需要求出逆序对数为奇数的排列数量减去偶数的排列数量即可。而这个值就是把这个矩阵看成一个行列式求值得到的结果。

解行列式可以使用消元法。由于行列式满足将某一行乘某个数加到另一行上结果不变的特点，所以可以类似高斯消元法进行消元。又由于这个矩阵的特殊性，可以保证在消元的时候每一行的右端点不变只有左端点右移，每一行一直都是连续的一段1。

具体做法：将每个左端点下所有的行的右端点用堆维护。从1到n依次扫描左端点，并将所有行减去右端点最小的那一行，即将堆删除最小元素后合并到后面的一个堆中。使用可并堆可以将这部分的复杂度做到$O(n\log n)$。

做完这一步以后就变成了n个左端点不相同的行。那么通过相互消元，每行的右端点会和左端点相同。也就是矩阵中只有n个1。那么这个行列式的值就是最终的值，只需要统计这些1的逆序对数即可算出最终答案。

时间复杂度：$O(n\log n)$

空间复杂度：$O(n)$

1.8 APRIL15 LPARTY

Description

给定一个n维空间中的$2*2*2*\cdots*2$的高维立方体，一开始每个小块都是白色的，每次可以选取一个子立方体将它全部染成黑色，代价为这个立方体在某一维上长度为1的维的个数。

给定目标状态，求最小代价。

$n \leq 5$

Analysis

这是一个最小带权集合覆盖问题，无法在多项式时间内解决，那么不妨考虑使用搜索。

一个显然的做法是：直接搜所有3^n的子立方体取或不取。但是这样显然是无法通过的，于是就需要加一些剪枝：

- 那些覆盖非法点的子立方体显然可以直接排除。
- 假如两个子立方体都合法，那么小的显然无用（范围小花费高）
- 搜索时假如发现后面所有都选上也无法满足了就退出。
- 搜索时假如发现这一个能覆盖的之前都覆盖了就不需要再加了。
- 最优性剪枝：如果当前代价已经比当前最优解大则退出。

这样即可通过本题。

时间复杂度：$O(2^{3^n})$

空间复杂度：$O(2^n)$

1.9 MARCH15 RNG

Description

给定一个序列A_i，满足$A_i = A_{i-1} * C_1 + A_{i-2} * C_2 + \cdots + A_{i-k} * C_k \ for \ i > k$。给出$A_1 \cdots A_k$和$C_1 \cdots C_k$，求$A_n$。所有数据模104857601。

$k \le 30000$,$n \le 10^{18}$。

Analysis

注意到我们并不需要直接求出A_N，而只需要将A_N表示成$\Sigma V_i \times C_i$的形式即可。那么问题就是如何求出这些系数V_i。

考虑暴力做法：每次将当前i最大的A_i替换成$\Sigma_j A_{i-j} * C_j$。这样的复杂度是$O(n)$的。注意到这么做与多项式取模非常相似。经过观察发现，V_i数组其实就是x^n模$x^k - C_1 * x^{k-1} - \cdots - C_k * x^0$的值。

由于多项式乘法、多项式取模都可以在$O(k \log k)$的时间内完成，那么只需要类似快速幂的做法倍增算出x^n即可。

时间复杂度：$O(k \log k \log n)$

空间复杂度：$O(k)$

1.10 MARCH15 TREECNT2

Description

给定一棵n个点的树，每个点的权值为a_i。再给出m个操作，每个操作会将一条边的权值修改为某个数，然后询问树上gcd为1的链有多少条。

$n \le 100000$,$m \le 100$,$a_i \le 10^6$。

Analysis

直接统计gcd为1的链并不方便，所以可以先算出gcd为每个值的倍数的链的数量，然后使用容斥即可计算出最终答案。

那么枚举gcd之后，只有那些是这个值的倍数的边会被使用到。首先将不在修改中的边使用并查集合在一起，然后对于每个询问，将修改中的边

加入然后统计答案（假如一个修改都没有涉及就可以不用枚举修改直接加入总答案）。由于每次询问后要复原并查集，所以需要使用可撤销的并查集，复杂度为$O(\log n)$而不是$O(\alpha(n))$

注意到一条边只会被用在X个gcd值中，也只有mX个gcd值需要修改，所以复杂度并不会在a_i上乘任何值。

时间复杂度：$O(a_i + (Q^2 + n)X \log n)$

空间复杂度：$O(nX)$,X为因子最多的a_i的因子数。

1.11 FEB15 CUSTPRIM

Description

定义三元组的乘法

def multiply((a1,b1,c1), (a2,b2,c2)):

s = (a1a2 + b1b2 + c1c2) + (a1b2 + b1a2) + (c1 + c2)

t = floor[s/2] + 16(c1 + c2) - c1c2

A = (t - 2(a1b2 + b1a2) - (a1c2 + c1a2) + 33(a1 + a2)+ (b1b2 - a1a2))

B = (t - 5(a1b2 + b1a2) - (c1b2 + b1c2) + 33(b1 + b2)+ (2b1b2 + 4a1a2))

if s is even: return (A-540,B-540,24)

else: return (A-533,B-533,11)

定义zero：若x*任何y=0，则称x是zero

定义单位元，若x*任何y=y，则称x是单位元

定义质数，若x不是zero且不能分解成两个非单位元的乘积，则称x是质数

给定一个三元组，问它是不是质数

$a, b \le 10^7$

Analysis

经过转化以后可以转化为判断一个数是不是质数的问题。使用miller-rabin判断即可。

时间复杂度：$O(\log(a+b))$

空间复杂度：$O(1)$

1.12 FEB15 DEVLOCK

Description

求有多少不超过N位十进制数是P的倍数且每位之和小于等于M，答案对998244353取模。

$n \le 10^9, P \le 16, M \le 15000$

Analysis

首先考虑每一位i，假如10^i余数一样那么它们就是完全相同的并没有区别。那么首先求出10^i模P余数为p的分别有几个。那么可以使用分治FFT求出只用某个余数的位，数位和为某个数的方案数。注意到余数是会循环的，也就是说个数最多只需要求3种而不是p种，所以复杂度是$O(M\log M\log N)$。

那么接下来要把这些多项式合并在一起。考虑$dp_{i,j}$表示使用前i个余数，总的余数是j的多项式，那么在转移的时候将i这个余数的多项式按照余数和拆成p 个多项式$A_{i,j}$进行转移即可。即：$dp_{i,j} = \Sigma_k dp_{i-1,k} * A_{i,(j-k) \bmod p}$。

注意这里每个多项式要乘p次，所以需要先DFT然后最后IDFT这样可以省下一个log。

时间复杂度：$O(M\log M(P^2+\log N))$

空间复杂度：$O(PM)$

1.13 JAN15 XRQRS

Description

给定一个初始为空的序列和n个操作，每个操作有可能是：

- 在序列最后插入数x

- 求区间内的值y使得$x\ xor\ y$最大。
- 删除最后x个元素
- 统计区间内比x小的元素个数
- 找区间内k小值

$n \le 5 * 10^5$

Analysis

第二种操作是Trie树的经典操作，第四种和第五种则是线段树的经典操作（当然也可以用Trie实现），为了支持区间查询和插入删除，只需要维护一个函数式Trie 树即可。

时间复杂度：$O(n \log n)$

空间复杂度：$O(n \log n)$

1.14 JAN15 RANKA

Description

在一个9*9的棋盘上下围棋，规定不能出现重复局面（即棋盘相同且当前下棋的人相同），轮到一个人时这个人可以不下，求一种下10000步的方案。

Analysis

注意到可以不下这个条件，我们可以让一个人铺满棋盘，然后另一个人下剩下的唯一一个点把另一个人吃了，然后双方互换角色继续铺满棋盘，这样大概可以重复81*81次（因为刚落子吃掉的局面不能重复），还不太够。所以可以让一个人铺得只剩两个位置再让另一个人吃，这样就能轻松达到10000步了。

时间复杂度：$O(1)$

空间复杂度：$O(1)$

2 传统题2014

2.1 DEC14 DIVIDEN

Description

给定一个x度角，要求只使用两个基本操作，将它分成x个1度角。

操作1：在已知的两点x,y之间连线；操作2：以一个已知点为圆心，以某两个已知点的间距为半径画圆。

所有直线和圆的交点均会变为已知点。

无解输出NO

Analysis

由于3度角是可以尺规作图画出来的，但是1度不行。所以加入x是3的倍数那么就无解。否则只需要构造出3度角然后相减得出一度角问题就解决了。

由于正五边形是可以尺规作图做出的，这样就获得了72度，将它折半便获得了36度，再和30度角相减再折半就是3度了。

时间复杂度：$O(1)$

空间复杂度：$O(1)$

2.2 DEC14 RIN

Description

现在需要为n个数a_i各确定一个介于$1 \sim m$之间的数，给出第i个数为j的收益$f(i,j)$，若$f(i,j) = -1$则表示不能选择。

再给出k个限制，每个限制形如$a_x < a_y (x \neq y)$。求最大收益。

$1 \leq m, n \leq 100$,$0 \leq k \leq 100$,$-1 \leq f(i,j) \leq 100$

Analysis

题目给出了$f(i,j)$的上界，所以可以把收益$f(i,j)$转变成固定的100收

益和造成的$100 - f(i,j)$点亏损。

由于需要保证每个位置只能有一个值，即每个点的取值建出来的割边在最小割中只能出现一个，所以需要将同一个点建出来的边串联起来。

那么考虑一种最小割模型：对于n个位置各建出m个点，并将它们连成一条线，割掉点(i,j)的入边代表选择$a_i = j$，所以权值就是$100 - f(i,j)$（当$f(i,j) = -1$时连无穷大的边表示不能被割掉）。容易看出这样建出来的模型每一行最小割中只会被割掉一条边，满足题设。

那么考虑如何加入限制：即i在第k条边割掉，j割掉的位置不能小于等于k。那么可以对于这样的每一组i,j都加上m条无穷大的边，分别从(i,k)连向$(j,k+1)$。当然还要禁止j变量为1，所以要从源向第一个点连无穷大的边。

跑一遍最小割即可解决。

时间复杂度：$O(n^2m)$

空间复杂度：$O(nm)$

2.3 NOV14 FNCS

Description

给定N个数a_i，N个函数f_i，每个函数表示一段区间的a_i的和（即$f_i = \Sigma_{x=l_i}^{r_i} a_x$，$f_i$的值会根据$a_i$的值的改变而改变）。有$N$ 个操作，每个操作可以：

- 修改一个数（a_i）
- 询问$\Sigma_{x=l}^{r} f_x$

$N \leq 10^5$

Analysis

先将操作分块，即每$H(N)$次操作重建一次，每次重建的时候$O(N)$算出原数组的前缀和并$O(N)$求出每个f_i的值，复杂度$O(N \times N/H(N))$。

接下来只需要计算每个操作对块内询问的贡献。由于询问是可减的，可以把每个询问拆成前缀询问，然后把每个前缀询问按长度排序。

排完序后就可以按右端点递增扫过来并维护一个支持$O(\sqrt{N})$区间加$O(1)$单点查询的数据结构，这个数据结构在之前提到过：把a_i每隔$O(H(N))$个分为$O(N/H(N))$块，每次修改只需要修改$O(N/H(N))$块和$O(H(N))$个单点，查询只需要把块内答案和单点答案相加即可。然后对于每个询问枚举和它在同一块中并且在它前面的所有$O(H(N))$个修改，由于可以$O(1)$查询某个位置对那个询问的影响，就可以在$O(N \times H(N))$ 的时间内算出这一部分的贡献。

时间复杂度：$O(N\sqrt{N})$

空间复杂度：$O(N)$。

2.4 NOV14 SEAORD

Description

给定n个任务和两台电脑，每个任务需要在电脑A上连续运行a_i秒在电脑B上连续运行b_i秒，且同一个任务不能同时在两台电脑上运行，求将所有任务都运行完最少需要多少时间并给出方案。

$n \leq 2 * 10^5$

Analysis

容易发现答案至少为$\max(\Sigma a_i, \Sigma b_i, a_i + b_i)$，那么问题的关键在于如何把所有任务都塞到这个时间段内。

容易发现当n比较大的时候非常不容易产生碰撞，当n比较小的时候排列方式又非常少，所以直接多次随机某一边的执行顺序然后贪心即可。

时间复杂度：$O(n)$

空间复杂度：$O(n)$

2.5 OCT14 TRIPS

Description

给定一棵N个点的树，每条边长度是1或2。有N个询问，每次给出x,y,z，要求找出尽量少的点$a_1 \cdots a_n$使得$dis(x,a_1), dis(a_1,a_2), \cdots, dis(a_n,y)$都不超过$z$（这里$dis(i,j)$表示树上$i$点和$j$点的距离）。

$N \leq 10^5$

Analysis

考虑暴力做法：每次二分下一个a_i的位置，这样是$O(N\log N)$的，因为a_i最多会有$O(N)$个。但是注意到当$z > H(N)$的时候a_i最多只有$O(N/H(N))$个可以直接暴力，而$z <= H(N)$的情况只有不超过$H(N)$种，假如把这些z的答案预处理出来，那么问题就解决了。

那么可以对每一个比较小的的z，都预处理出每个点向根走z长度（假如没有距离为z的点那么就选取距离为$z-1$的点）到的是哪个点，并倍增出每个点向上走2^k 步（每步长度为z）到的是那个点。这样需要$O(N \times H(N)\log N)$ 的时间和空间。询问的时候可以像正常的倍增算法一样向上走到再向上走就超过LCA为止，最后多出来的那一段可以暴力在$O(H(N))$的时间内走完。

取$H(N) = \sqrt{N}$就可以在$O(N\sqrt{N}\log N)$的时间和$O(N\sqrt{N}\log N)$的空间下解决这道题。

时间复杂度：$O(N\sqrt{N}\log N)$

空间复杂度：$O(N\sqrt{N}\log N)$

2.6 OCT14 BTREE

Description

给定一棵树，边权均为1。

给出Q组询问，每次在k_i个点上放置警卫，警卫可以保护与他所处的位置距离不超过r_j的点。问有多少个点至少能得到一个警卫的保护。

$1 \leq n \leq 50000$,$1 \leq Q \leq 50000$,$1 \leq r_j \leq n$,$\Sigma k_i \leq 500000$

Analysis

假如每个询问只有一个警卫，那么这个问题可以使用点分治解决。因为题目没有要求强制在线，我们可以使用离线算法，记录下每个点有哪些长度的询问。

进行点分治的时候通过记下每个子树的每个深度的点各有多少个，以及除了某个子树以外的每个深度的点的个数，那么就可以很方便地在$O(size)$的时间内统计出经过这个点的答案对每个询问的贡献。

在有多个点的情况下就不是那么好处理了。

首先为了保证询问只和Σk_i有关，需要建出一棵包含询问点的虚树。

接着考虑虚树上的每条边，设它的长度为l，端点为a,b，a的控制范围为p_a，b的为p_b且$p_a \le p_b$。

假如$p_a + l > p_b$，那么这两个警卫会有重叠部分，而重叠的那部分刚好可以转化为一个新的警卫（只不过这个警卫的贡献是负的）。因为这条链上一定存在一个点（有可能在一条边的重点上）使得两个警卫到这个点后能管得剩余距离相同，那么这一段就是完全重复的。于是这就可以通过新建一个询问并在点分治中一起解决。

假如$p_a + l \le p_b$，那么a警卫管制的区域会完全地被b警卫覆盖，那么就可以直接将p_a替换为$p_b - l$，然后用新的p_a接着更新其他点。

但是在这里暴力更新是不行的，所以要通过一个虚树上的DFS算出每个点被哪个点完全地管制了。在计算出最后的答案后就可以在每条边上新建一个负询问来去除重复部分。

添加完新的询问以后每个询问就变成了许多个$k_i = 1$的询问的和（或差），套用$k_i = 1$时的点分治做法即可。

时间复杂度：$O(\Sigma k_i \log n + n \log n)$

空间复杂度：$O(n \log n)$

2.7 SEPT14 FIBTREE

Description

给定一棵n个点的树，有n个操作，每个操作有可能是：

- 将一条路径上的点分别加上1,2,3,5,8...的斐波那契数。

- 查询当x为根时，以y为根的子树中的权值和。

- 查询x到y路径上的权值和。

- 将所有节点的权值还原到第x个操作之后的状态。

所有数值模10^9+9

$n \leq 10^5$

Analysis

首先随便取一个根，将树轻重链剖分一遍，那么二三两个操作就分别变成了区间查询和$O(\log n)$个区间查询（二操作的查询只可能是子树或者子树的补），一操作就变成了$O(\log n)$个区间加斐波那契数。那么这个问题就变成了序列问题。

注意到$\sqrt{5}$在模10^9+9的情况下是存在的，又由于$fib_i = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^i + (\frac{1-\sqrt{5}}{2})^i)$，所以区间加斐波那契数可以转化为区间加等比数列。这样就可以用线段树打标记来维护。

实现第四个操作只需要将线段树可持久化即可。

时间复杂度：$O(n\log^2 n)$

空间复杂度：$O(n\log^2 n)$

2.8 SEPT14 QRECT

Description

给定一个二维平面和n次操作，每次操作可以：

- 新建一个矩形

- 删除一个矩形

- 给定一个新矩形，查询有多少个矩形和这个矩形有公共面积

$n \leq 10^5$

Analysis

查询的时候使用容斥进行统计，那么需要统计的数量其实就是所有矩形减去上、下、左、右的矩形再加上右上、右下、左上、左下的矩形。也就是说现在只需要求一个区域内有多少个矩形。

以右上为例，一个矩形右上方的矩形个数其实就是右上方的矩形左下角个数。于是问题就转化为了一维数点和二维数点问题。

二维数点问题可以使用CDQ分治离线完成或者树套树在线完成。使用CDQ分治可以只使用$O(n \log n)$的空间。

时间复杂度：$O(n \log^2 n)$

空间复杂度：$O(n \log^2 n)$

2.9 AUG14 SIGFIB

Description

给定m, n，求$\Sigma_{x+y+z=n} 6xyz * fib_x * fib_y * fib_z \bmod m$。

$n \leq 10^{18}, m \leq 10^5$

Analysis

可以推出这个值关于n的母函数，直接计算第n项即可。

其中母函数中需要用到斐波那契数的第$O(n)$项，使用矩阵快速幂即可。母函数中还需要用到组合数不过有一维很小可以直接暴力做除法。

时间复杂度：$O(\log n)$

空间复杂度：$O(1)$

2.10 AUG14 PUSHFLOW

Description

给定一棵n个点的点仙人掌，每条边有一个流量限制，给出m个询问，每个询问可以：

- 修改一条边的流量
- 询问两点之间的最大流

$n \le 10^5, m \le 2 * 10^5$

Analysis

将每一个环上的最短边都拆出来容易发现最大流中这个环内的这一半的其他边的边权并不重要。于是只需要将每个最短边都在对应的链上加上自己的边权，查询时直接链上取min即可。

修改的时候由于最短边会发生变化，所以需要找到最短的那条边重新建树。也就是说需要支持link-cut操作。

以上操作LCT均可实现，使用LCT维护即可。

时间复杂度：$O((m+n)\log n)$

空间复杂度：$O(n)$

2.11 JULY14 SEAEQ

Description

对于两个全排列a,b，定义$F_E(a,b)$为(l,r)的对数满足$1 \le l \le r \le n$，$a_l \cdots a_r$包含不超过E个逆序对且$a_l \cdots a_r$和$b_l \cdots b_r$ 离散后相同。

给定n,E，求$\Sigma_{length(a)=length(b)=n} F_E(a,b)$

$n \le 500$,$E \le 5000000$

Analysis

首先我们可以将枚举排列转化为枚举区间计算贡献。那么区间的贡献显然和位置无关，只和长度有关。那么问题就转化为了求一个长度为i的区间对答案造成了多少贡献。

那么首先需要求出逆序对数不超过E的长度为i的全排列个数。这个可以通过一个dp得到：$dp_{i,j}$表示长度为i逆序对数为j的有多少个。复杂度为$O(n^3)$。

容易发现不管全排列是怎么样的，对答案的贡献都是相同的，即$\binom{i}{n} * (n-i)!$，直接计算即可。

时间复杂度：$O(n^3)$

空间复杂度：$O(n^3)$

2.12 JULY14 GNUM

Description

给定两个数组a,b,需要取出尽可能多的数对$(i,j),(p,q)$，且满足$b_i < a_j, b_p > a_q, \gcd(b_i,a_j) \neq 1, \gcd(b_p,a_q) \neq 1, \gcd(\gcd(b_i,a_j),\gcd(b_p,a_q)) \neq 1$，并且每对$(i,j)$不重复，每对$(p,q)$不重复。求最多能取出几对。

$n \leq 400$,$a_i, b_i \leq 10^9$

Analysis

可以把每对数(u,v)抽象成一个点，可以同时选的点之间连边。由于$b_i < a_j, b_p > a_q$，所以这个图是一个二分图，而本问题就是一个二分图匹配问题。

直接暴力建图的复杂度是$O(n^2)$个点，$O(n^4)$条边的匹配，显然会超时。那么需要加一些优化。

第一种优化：由于每对点是什么并不重要，只需要记下每对点的gcd即可。那么将gcd的结果去重，最多只有$n\sqrt{n}$种，那么就是$O(n\sqrt{n})$个点，$O(n^3)$条边，当然这种做法的常数非常非常小，几乎不能卡到上界。

第二种优化：由于匹配的条件就是$\gcd(b_i,a_j,b_p,a_q) \neq 1$，也就是只要对于同一个质数，两边的数对都是它的倍数就可以增加1的答案。那么将出现过的质数都新建一个点，两边的数对都向中间连边做网络流即可，这样做就是$O(n^2)$个点，$O(n^2 \log n)$条边。

时间复杂度：$O(n^3 \log^{1.5} n)$

空间复杂度：$O(n^2 \log n)$

2.13 JUNE14 SEAARC

Description

给定一个长度为n的序列A，求所有的四元组$a < b < c < d$满足$A[a] = A[c], A[b] = A[d]$。答案模$10^9 + 7$。

$n \leq 100000$

Analysis

这题的信息看起来不是很好统计，那么不妨考虑分块算法。

将所有元素分成两类：出现次数大于$\sqrt{n}$的和不大于$\sqrt{n}$的。不妨称他们为多元素和少元素。那么只需要统计三个信息：多多之间的个数，少少之间和多少之间。

多多之间：枚举两个多元素，使用two-pointer扫一遍数组，扫的过程中使用类似dp转移的方法维护$a, ab, aba, abab, b, ba, bab, baba$的数量。由于多元素最多只有$\sqrt{n}$个所以复杂度是$O(n\sqrt{n})$。

少少之间：将所有少元素按顺序排序，扫过来的同时维护所有值相同的点对跨越的区间，使用树状数组求出某一个点对对答案的贡献。由于一个少元素最多只会连出$\sqrt{n}$条边，复杂度为$O(n\sqrt{n}\log n)$。

多少之间：类似多多之间，不同的是枚举两个块，然后在大块里二分出小块每个元素的位置。复杂度为$O(n\sqrt{n}\log n)$。

时间复杂度：$O(n\sqrt{n}\log n)$

空间复杂度：$O(n)$

2.14 JUNE14 TWOCOMP

Description

给定一棵不超过100000个点的树，上面有$2n$条路径，分为两种颜色，每条路径有一个权值。需要选择权值和尽可能大的路径使得任意两条颜色不同的路径不相交。

$n \leq 700$。

Analysis

容易发现题目要求的是二分图最大权独立集，这个问题可以使用网络流解决。那么问题的关键就是如何建图，即快速判断两条路径是否相交。

以一个点为根建出有根树，那么对于每条路径，使用两个点的LCA将路径切分成两条连续的路径，容易发现四条路径中两条路径相交必定是某一条路径的端点在另一条路径上，不可能是端点分离内部相交。于是枚举路径对判断端点是否在路径内部即可。

时间复杂度：$O(n^3)$

空间复杂度：$O(n^2)$

2.15 MAY14 ANUDTQ

Description

给定一棵有n个点的树和m个操作。有四种操作：

- 给定一个节点x，生成一个新节点作为这个节点的孩子，新节点的编号将是从未使用的正整数中最小的一个，节点的值将给定。
- 给定一个节点x，将以x为根的子树中所有节点的value加上给定值。
- 给定一个节点x，将以x为根的子树中所有节点删除。
- 给定一个节点x，询问以x为根的子树中所有节点的value值的和。

强制在线。

$n \leq 50000$。

Analysis

首先求出这棵树的DFS序，并用一棵平衡树维护。那么四个操作就分别变成了：插入一个点，区间加一个值，删一个区间，区间和查询。这些都是基础的平衡树操作，使用任意一种平衡树维护即可。

时间复杂度：$O(n \log n)$

空间复杂度：$O(n)$

2.16 MAY14 SEINC

Description

给定两个序列A,B，每次操作可以将序列A中的一个区间$+1$，求最少需要多少此操作才能使A和B在模4意义下相同。

$n \leq 10^6$。

Analysis

首先将B减去A，那么要求的就是从全0数列变到一个数列的次数。

将这个数列进行差分，那么一次操作就变成了选一对数(i,j)满足$i \leq j$然后将a_i++,a_j--。

将差分后的数列每一项模4，使得每一项都为$0 \sim 3$，那么题目要求其实就变成了将某些项-4，满足任意长前缀和不小于0且正数项之和最小。注意到将3,2,1减4带来的贡献为3,2,1但对于前缀和的损失却是一样的。所以应该优先选择3。对于3，应该优先选择更后面的。所以只需要从后往前扫3遍即可。

时间复杂度：$O(n)$

空间复杂度：$O(n)$

2.17 APRIL14 GERALD08

Description

有一棵树，每条边分别属于一个人，两个人轮流操作，每次删掉一条属于自己的边，然后将与根不连通的子树删除。不能操作就输。求在某个人先手的情况下谁能赢。

$n \leq 100000$。

Analysis

将每棵子树使用Surreal Number表示，那么假如能够计算出递推式，求出根结点的Surreal Number以后判断正负即可。

打表或推导后能得出公式：对于一个孩子i，如果连接它的那条边属于第一个人，那么令x为满足$f_i + x > 1$的最小正整数，i的贡献就是$\frac{f_i+x}{2^{x-1}}$。属于第二个人转移类似。

容易发现这样Surreal Number可能有$O(n)$位小数，所以需要使用高精度。

可以使用set维护小数中1的位置，使用启发式合并即可保证复杂度。

时间复杂度：$O(n\log^2 n)$

空间复杂度：$O(n\log n)$

2.18 MARCH14 GERALD07

Description

给定一张n个点m条边的无边权的无向图。给出n个询问，每次询问使用编号为L_i到R_i的区间内的边构成的子图中连通块的个数。

$n, m \leq 200000$。

Analysis

首先考虑算出所有右边界为n的询问的答案：从大到小加入边，使用并查集维护即可。

那么假如需要求出右边界不为n的答案，就需要删除不能使用的最大的那些边。由于加入一条边时最多只会有一条边变成无用的，所以在加边的时候用LCT维护最小时间生成森林即可求出每次加边时该删除哪条边。那么询问其实就是一个二维数点，使用可持久化线段树维护即可。

时间复杂度：$O((n+m)\log(n+m))$

空间复杂度：$O(n+m)$

2.19 MARCH14 STREETTA

Description

有两个长为n的数组A, B，初始时A全为0，B全为-inf，有m个操作，

每个操作为：

- 在A数组中的一段上加等差数列。
- 在B数组中的一段上分别和等差数列的每一项取max
- 询问$A_x + B_x$

$n \le 10^9, m \le 300000$

Analysis

第一个操作很容易实现，在线段树每个节点上打一个首项为xxx差为xxx的等差数列的标记，标记是可以直接合并的而且不用下传。

容易发现任意x个等差数列只会分出x段区间，那么在线段树的每个节点上打一个标记表示这整个节点都被这个等差数列覆盖，当两个等差数列相交时，找出交点所在的那一侧，那么另一侧是只由一个等差数列控制的，递归操作交点那一侧即可。每次操作需要修改$O(\log n)$个结点，每个结点下传需要$O(\log n)$的时间。

时间复杂度：$O(m \log^2 n)$

空间复杂度：$O(m \log n)$

2.20 FEB14 DAGCH

Description

给定一张n个点m条边的有向图，并给出它们的DFS序。一个节点x是另一个节点y的半必经点假如x是能只通过DFS标号大于y的标号的点到达y的编号最小的点。求每个点是多少个点的半必经点。

$n \le 10^5$。保证1能到达所有点。

Analysis

注意到这里半必经点的定义就是Dominator Tree中的半必经点的定义，直接求出Dominator Tree统计即可。

时间复杂度：$O(n\alpha(n))$

空间复杂度：$O(n)$

2.21 FEB14 COT5

Description

有一棵空的Treap，给出n个操作，每个操作可以

- 插入一个关键字和权值都给定的点
- 删除一个点
- 求两个点之间的距离

$n \le 2 * 10^5$。

Analysis

直接维护Treap形态显然是不现实的，所以考虑维护按照关键字排序以后的权值数组。

由于两个点的距离就是两个点深度之和再减去LCA的深度的两倍，而LCA就是这两个点之间权值最大的点，所以只需要支持查询一个点的深度这一个操作即可。

容易发现一个点是另一个点的祖先当且仅当它们之间没有更大的点。那么考虑如何求出祖先个数。

那么在查询的时候其实是需要快速求出这样一个信息：在一个子树中以从左到右/从右到左的顺序查询上一个数大小是k的答案。容易发现大多数查询我们需要的是k等于这个子树的另一半的子树中的最大值。那么在修改的时候实时维护k为这个值时的答案，那么一个任意k的查询就可以在$O(\log n)$的时间内求出了。

由于一次询问的时候会调用$O(\log n)$次任意k的查询，而剩下的信息可以直接使用维护的信息，所以复杂度为$O(\log^2 n)$。

时间复杂度：$O(n \log^2 n)$

空间复杂度：$O(n)$

2.22 JAN14 CNTDSETS

Description

在一个高维空间中，定义两个点的距离为各维坐标差的最大值，定义点集直径为距离最大的两个点的距离，定义两个点集相等为它们之间可以通过平移得到。

给出n, m，求n维空间中直径为m的点集个数。

$n \leq 1000, m \leq 10^9$

Analysis

可以先求出直径不超过m的点集个数，然后和$m-1$的相减即可求出答案。直径是m其实就是每个点每一维坐标都在$1 \sim m$之间。

为了防止点集重复，不仅需要保证每个点每一维坐标都在$1 \sim m$之间，还需要保证每一维点坐标的最小值都是0。那么使用容斥，枚举至少有多少维坐标的最小值是0的答案分别计算。

时间复杂度：$O(n^2)$

空间复杂度：$O(n^2)$

2.23 JAN14 TAPAIR

Description

给定一张n个点m条边的无向图，问有多少种方法删掉两条边使得这张图不连通。

$n \leq 10^5, m \leq 3 * 10^5$

Analysis

先求出这个图的DFS树，那么要选出一个边集使得这个图不连通必须满足没有边连接树边的两侧。定义一条非树边覆盖一条树边为这条树边在非树边端点的链上。那么一个边集使得这个图不连通等价于存在一个这个边集的子集满足每条非树边都经过偶数条树边。

直接查询的复杂度过高，可以给每条非树边一个随机权值，使用异或和是否为0来判断是否每条边都经过偶数次。

时间复杂度：$O(m \log m)$

空间复杂度：$O(m)$

3 传统题2013

3.1 DEC13 REALSET

Description

给定一个长度为n的序列$A_0 \cdots A_{n-1}$问是否存在一个长度为n的序列$B_0 \cdots B_{n-1}$满足：

- 至少存在一个$0 \le i \le n-1$满足$B_i \ne 0$
- 对于任意$0 \le j \le n-1$满足$\Sigma_{i=0}^{n-1} A_i * B_{(i+j) \bmod n} = 0$

$n \le 30000$

Analysis

问题等价于判断一个循环矩阵是否满秩，同时等价于$\gcd(f(x), x^n - 1)$的次数是0，其中$f(x) = \Sigma A_i x^i$。

使用$\phi_i(n)$表示分圆多项式，那么问题就等价于是否存在一个d满足$d|n$且$\phi_d(x)|f(x)$，同时$\phi_d(x)|f(x)$等价于$x^d - 1|f(x) \prod_{p\ is\ prime, p|d} (x^{\frac{d}{p}} - 1)$。而这个非常容易判断，直接暴力即可。

时间复杂度：$O(n\tau(n) \log n)$

空间复杂度：$O(n)$

3.2 DEC13 QTREE6

Description

给定一棵n个点的树，点分为黑色和白色，每次可以修改一个点的颜

色，或者查询一个点所在的和它相同颜色的点组成的连通块大小。

$n \le 100000$

Analysis

查询操作可以分为两步：首先找出一个点连续向上走相同颜色的结点能走到的最高点；然后维护不算某个点的父亲边的连通块大小，查询那个点的值即可。

第一步其实就是每次倍增，直到一段区间内不全是这个颜色就停止。也就是说需要查询某个点到根路径上的黑点个数。这是单点修改-到根路径查询，可以变成子树修改- 单点查询，DFS序处理即可。

第二步可以在每次更新的时候都将答案改变的答案修改一下，因为改变答案的点就是这个点到第一步求出来的那个点之间的路径上的点。但是暴力计算改变就需要枚举儿子节点然后加起来无法处理。所以对于每个点需要额外维护这个点是黑的答案和是白的答案，那么就能$O(1)$算出这个点的答案变化了。由于DFS序无法直接修改一条链，可以使用差分将答案变为修改两个点。

时间复杂度：$O(n \log^2 n)$

空间复杂度：$O(n)$

3.3 NOV13 MONOPLOY

Description

给定一棵n个结点的有根树，每个结点的权值都不同。

给出m个操作，每个操作可以将一个点到根的路径上的点的权值都改成和这个点相同，或者询问一个子树内到根路径上不同权值数的平均值。

$n, m \le 200000$

Analysis

注意到一个点到根的不同权值数就是路径上和自己的父亲权值不同的点的个数。于是直接维护哪些点相同哪些点不同即可。

容易发现修改操作和Link-cut Trees的access操作非常相似，可以使用LCT的复杂度分析来说明直接暴力更改所有的分界点的操作次数是$O(n \log n)$的。

那么对于每个点维护这个点到根的答案，那么每次修改分界点相当于子树修改，询问相当于子树和查询。于是使用DFS序维护即可变成序列操作，然后使用树状数组解决。

时间复杂度：$O(m \log^2 n)$

空间复杂度：$O(n)$

3.4 NOV13 QPOINT

Description

给定平面上n个简单多边形，保证没有一个点同时在两个多边形内部。

然后给出m个询问，每次询问一个点在哪个多边形内或者哪个都不是。

强制在线。

$n \leq 10^5$，多边形边数之和不超过$3 * 10^5$，$m \leq 10^5$。

Analysis

使用扫描线，将所有点x坐标作为关键坐标扫过来，那么相邻两个坐标之间边的顺序是不变的，只需要插入删除一条边即可，所以可以使用平衡树维护。在插入删除的同时维护每一条边在多边形的左边还是右边，那么询问时只需要查出左边第一条边，然后看看这条边是不是属于左边界即可。

由于题目限制了强制在线，将平衡树可持久化即可。

时间复杂度：$O((m+n) \log n)$

空间复杂度：$O(n \log n)$

3.5 OCT13 FN

Description

定义F_m为斐波那契数列的第m项，给定n, c，求最小的m满足$F_m \bmod n = c$。

$c < n \leq 2 * 10^9$,n为质数且$n \bmod 10$为完全平方数。

Analysis

由于$F_n = \frac{1}{\sqrt{5}} \times ((\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n)$，设$x = \frac{1+\sqrt{5}}{2}$，那么当$n$为偶数时，$x^n = \frac{c \pm \sqrt{c^2+4}}{2}$，$n$为奇数时类似。

那么使用cipolla求出$c^2 + 4$的二次剩余，然后使用大步小步求出n即可。

时间复杂度：$O(\sqrt{n})$

空间复杂度：$O(\sqrt{n})$

3.6 SEPT13 TWOROADS

Description

给定平面上的n个点，要求选出两条直线，使得每个点到两条直线上的点中最近的那个点的距离之和最小。

$n \leq 100$

Analysis

注意到每个点归属于哪条直线的区域分界线是两条直线的角平分线，那么可以枚举每个点的归属，使得两条直线互不相干，然后对于每一部分使用最小二乘法解出答案。

注意到最优的情况一定是一条角平分线经过至少两个点，另一条经过至少一个点。暴力枚举是$O(n^4)$的，那么在枚举一条线以后将剩下点排序，然后按顺序扫过来是$O(1)$更新的，就可以做到$O(n^3 \log n)$了。

时间复杂度：$O(n^3 \log n)$

空间复杂度：$O(n)$

3.7 AUG13 PRIMEDST

Description

给定一棵n个点的树，求随机选取两个不同的点，它们之间距离为质数的概率是多少。

$n \leq 50000$

Analysis

容易发现质数这个条件并不能很好的利用，最终还是需要求出每个长度的路径的方案数。

那么使用点分治每次将经过当前选择的点的路径统计出来。可以枚举每个子树然后算出它和其他所有子树之间的贡献。

注意到贡献的式子就是一个多项式乘法，使用FFT进行加速可以做到每一层$O(n \log n)$，总复杂度$O(n \log^2 n)$。

时间复杂度：$O(n \log^2 n)$

空间复杂度：$O(n)$

3.8 AUG13 LYRC

Description

给定m个模板串和n个匹配串，求每个模板串在匹配串中出现的总次数。

所有串长总和不超过2500000。

Analysis

AC自动机可以解决一类多串匹配的题目。对于这道题，只需要建立出模板串的AC自动机，然后匹配串在上面跑并留下标记，最后再在fail树上求子树和即可。

本题是AC自动机的模板题，属于作业中最简单的题之一。

时间复杂度：$O(|S|)$,其中$|S|$为所有字符串的长度和。

空间复杂度：$O(|S|\Sigma)$

3.9 JULY13 RIVPILE

Description

给定平面上n个点，有m个大小的圆，每个圆有半径和费用。你需要给每个点上放上一个圆，使得连通$y=0$和$y=k$的情况下费用最小。

Analysis

将每一个点拆成m个点表示放哪种圆。暴力做法需要连$O(m^2n^2)$条边，是无法接受的，需要优化边数。

首先可以去掉那些被其他某一个圆完爆的圆，于是剩下的圆按照半径排序后费用也是递增的。显然可以将一个更小的圆补差价替换成更大的圆而对之前没有影响。于是对于n个点拆出来的点内部都连上边。然后对于每一对原来的点，只需要对于拆出来的每一个点都连出一条最小可以接触到的边即可。这样做边的数量为$O(n^2m)$，可以接受。

时间复杂度：$O(n^2m\log n\log m)$

空间复杂度：$O(n^2m)$

3.10 JUNE13 TKCONVEX

Description

给出n条木棍，要求从中取出$2m$条组成两个m边形。求一种方案或输出无解。

$n\le 1000, m\le 10$

Analysis

组成m边形的条件是最长的那条边的长度小于剩下的边之和。那么在

确定了两条最长的边之后剩下的边一定是尽量大。

假设两条最长的边间距超过m，那么显然就是取连续的$m-1$条边当剩下的边，这个可以$O(n)$求出。

假设间距不超过m，那么搜索这$2m-2$条边的边的归属即可。

时间复杂度：$O(n\binom{m}{2m})$

空间复杂度：$O(n)$

3.11 JUNE13 SPMATRIX

Description

给定一个$N*N$的整数矩阵A。这个矩阵如果满足以下条件就称它是特殊的：

- $A_{x,x} = 0\ for\ 1 \le x \le N$
- $A_{x,y} = A_{y,x} > 0\ for\ 1 \le x < y \le N$
- $A_{x,y} \le \max(A_{x,z}, A_{z,y})\ for\ 1 \le x, y, z \le N$
- $A_{x,y} \in 1, 2, \cdots, N-2\ for\ 1 \le x < y \le N$
- 任意$k \in 1, 2, \cdots, N-2$存在$x, y \in 1, 2, \cdots, N$满足$A_{x,y} = k$

给定n，求$n*n$的特殊矩阵个数。答案模10^9+7

$n \le 10^7$

Analysis

打出前几项的表之后可以通过OEIS获得答案的公式，只需要预处理出$1 \sim n$的阶乘和阶乘倒数和2^n的值就可以$O(1)$回答询问了。

$O(n)$处理阶乘倒数可以使用$inv_i = -inv_{mod\%i} * (mod/i)$来递推计算。

时间复杂度：$O(n)$

空间复杂度：$O(n)$

3.12 MAY13 QTREE

Description

给定一张n个点n条边的带权无向连通图，保证环长为奇数。定义两个点之间的路径为他们之间的最短路。给出m个询问，需要支持两种操作：

- 对一条路径上的点权取相反数
- 在一条路径上找出一条连续的路径使得权值和尽量大

$n \le 10^5$,$m \le 10^5$

Analysis

对环外的森林维护树链剖分，同时对环也开一个线段树单独维护，那么所有操作就都转化成了序列问题。

在序列上维护这个东西只需要支持打翻转标记并且维护六个值（正、负的左端连续、右端连续、中间最大）的线段树即可。

时间复杂度：$O(m \log^2 n)$

空间复杂度：$O(n)$

3.13 MARCH13 LECOINS

Description

有n种硬币，每种硬币有面值V_i和颜色C_i。给出m个询问，每个询问给出一个数x求组成总面值为x的钱最多能用多少种不同颜色的硬币。

$n \le 30, m, V_i \le 200000, x \le 10^{18}$

Analysis

假如能求出模某种硬币面值的情况下最小能够合出的那几个面值，那么大的面值自然能组合出来。那么随便选取一种硬币，将它的面值作为模数，求出不同余数下的最小能组合出的面值，那么为这个余数的更大的面值一定都能组合出来了。

那么用$dp_{i,j}$表示使用i种硬币，总面值模j下的最小面值，枚举颜色进行背包即可。

时间复杂度：$O(n^2m)$

空间复杂度：$O(nm)$

3.14 FEB13 ROC

Description

给出一条网格图上的封闭折线，折线有n行，保证直线内部任意两个同一y坐标的点之间的连线不经过直线。

在折线内部的每个90度角的地方有一个人，这些人围成了一个圈，任意两个相邻的人可以交换位置，花费的时间为他们之间的距离，但是在走的过程中不能走到折线外。

有m组询问，每组询问求两个人相遇的最短时间（只通过交换两个人达到）。

$n \leq 2500, m \leq 10000$

Analysis

容易发现一行最多只有两个人，所以人数是$O(n)$的。

那么首先需要在读入的时候预处理出所有人的位置关系和相邻两个人之间的距离。由于90度角一定会有人，那么两个人之间的距离超过曼哈顿距离的情况一定就是中间隔了一堵"墙"，而且这堵墙是没有凹陷的。也就是说只需要记下相邻两行之间的最左最右的坐标差，有人存在时直接调用记录的坐标差即可。

由于一个人不能被同时交换，那么相遇点只可能在两个人的中点位置。那么枚举那个位置然后取最小的即可。使用二分可以优化到$O(m \log n)$。

时间复杂度：$O(m \log n)$

空间复杂度：$O(m \log n)$

3.15 FEB13 QUERY

Description

给定一棵n个点的树，给出m个操作，每个操作可以：

- 路径加等差数列
- 查询路径和
- 将当前的树变成第x次操作后的形态

强制在线

$n, m \leq 100000$

Analysis

使用树剖可以将树链操作变为$O(\log n)$个序列操作。加等差数列可以在线段树上打永久标记来做到，而且标记显然是可以合并的。由于不需要下传标记，回溯操作可以直接使用函数式线段树解决。

时间复杂度：$O((m+n)\log^2 n)$

空间复杂度：$O((m+n)\log^2 n)$

3.16 JAN13 ANDOOR

Description

在一个$w \times h$的区域上，给定n个圆，求这些圆的并在区域内的周长。

$n \leq 5000$

Analysis

容易发现每个圆的贡献可以独立计算，就是说只需要计算每个圆有多少周长没被覆盖了即可。

那么枚举每个圆，将其他圆覆盖了它和边界覆盖了它的区间记下，然后排序扫一遍求并即可。

时间复杂度：$O(n^2)$

空间复杂度：$O(n)$

3.17 JAN13 CUCUMBER

Description

给定m个$n \times n$的矩阵A_i，对于一对数(x, y)，定义$B_{i,j} = \Sigma_{k=1}^{n} A_{x,i,k} A_{y,j,k}$，定义一个$1 \sim n$的排列是好的当且仅当至少存在一个$i$使得$B_{i,p_i}$是奇数。求有多少对数$(i, j)$有奇数个好的排列。

$n \le 60$,$m \le 8000$

Analysis

定义矩阵$C_{i,j} = (B_{i,j} + 1) \bmod 2$，那么在$C$的行列式中，每一个好的排列贡献就是0，剩下的是±1。特判掉$n = 1$的情况后，(x, y)是好的等价于$\det(C)$为奇数。那么暴力做法就是枚举(x, y)算出C的值，再消元求出$\det(C)$，复杂度为$O(n^3 m^2)$。

注意到假如在每个矩阵最后都补上一列1，那么$C = A_x \times A_y^T$。由Cauchy-Binet公式，设$A_{i,j}$为A_i删掉第j列的矩阵，则$\det(C) = \Sigma i = 1^{n+1} \det(A_{x,i}) \det(A_{y,i})$，也就是说只需要求出$A_{i,j}$的值即可。这个可以用消元解决。使用bitset优化可以加快速度。

时间复杂度：$O(n^3 m/w + m^2)$

空间复杂度：$O(n^3 m/w)$

4 传统题2012

4.1 DEC12 DIFTRIP

Description

给定一棵n个点的根为1号点的树，每个结点上有一个数字，这个数字等于这个结点的度数。

现在需要选出尽可能多的路径满足：

- 任意一条路径都是从根到某个点的路径的一部分。
- 任意两条路径上标注的数字按顺序排列后形成的字符串不同（长度不同或任意一个位置不同）。

$1 < n < 10^5$

Analysis

假如这道题的树形态是一条链，那么就是简单的子串数量统计问题，可以用后缀数据结构轻松解决。那么问题的关键就变成了如何将序列数据结构扩展到树上。

注意到这些串都是从根到某一个点的串的子串，那么只需要在树上建立出后缀自动机即可查询子串数量。

建立方法与对字符串建立后缀自动机类似。在字符串上建立时我们会对于每个前缀记录下代表这个串的那个节点，然后在插入下一个字符时在这个节点的基础上进行添加。在树上建立时同样记录下代表每个“前缀”（即根到一个点的一条路径）的节点，插入新点时在它的父亲的节点后插入即可。

在后缀自动机中查询子串数量是一个很基本的操作，只需要将每个点代表的子串数量相加（这个点的length减去它fail的length）。

时间复杂度：$O(n \log n)$

空间复杂度：$O(n \log n)$

4.2 DEC12 QPOLYSUM

Description

给定一个D次多项式$P(x)$的前$(D{+}1)$项$P(0) \mod M, P(1) \mod M, \cdots, P(D) \mod M$分别为$A_0, A_1, \cdots, A_D$，非负整数Q，正整数M,N，求$(P(0) \times Q^0 + P(1) \times Q^1 + \cdots + P(N-1) \times Q^{N-1}) \mod M$ 的值。

$1 < M < 10^{18}$,$0 \le Q < M$,$1 \le N < 10^{100000}$,$0 \le D < 20000$,$0 \le A_i < M$,$D \le 20000$,$N \le 10^{10^5}$,M不能被2至$D + 14$中的任意一个数整除

Analysis

令$G(n) = \Sigma_{i=0}^{n-1}P(i)Q^i$，通过数学归纳法可以得出$G(n) = Q^nF(n) - F(0)$，其中$F(x)$为一个次数为D的多项式。

由$G(n) = Q^nF(n) - F(0)$得出$G(n+1) - G(n) = P(n)Q^n = Q^{n+1}F(n+1) - Q^nF(n)$，即$P(n) = QF(n+1) - F(n)$，$F(n+1) = \frac{F(n)+P(n)}{Q}$。这样就得到了$F(n)$的递推式。

虽然我们不知道$F(n)$任意一项的值，但是可以根据递推式把$F(1), F(2), \cdots, F(d+1)$表示成关于$F(0)$的一次函数。

由于$F(n)$是一个次数为D的多项式，那么就满足$\Sigma_{i=0}^{d+1}(-1)^i\binom{d+1}{i}F(i) = 0$，于是利用这个方程就得到了一个关于$F(0)$的一次函数，解出$F(0)$即可。之后利用拉格朗日插值法即可算出$F(n)$。

但是注意到在递推和解方程的过程中会涉及除法运算，但在模M的情况下不一定有逆元。

最后方程中$F(0)$的系数为$\Sigma_{i=0}^{d+1}(-1)^i\binom{d+1}{i}Q^{-i} = (1-Q^{-1})^{d+1}$，所以需要用到$Q-1$的逆元。同时在递推式中也需要用到$Q$的逆元。那么假如在模$M$下没有$Q$或$Q-1$的逆元就会无法处理。

特判掉$Q=0$和$Q=1$的情况以后，容易想到可以将模数拆成几个小模数分别计算然后利用中国剩余定理合并。

那么令$M = m_1m_2m_3$，其中$(m_2m_3, Q) = 1, m_1|Q^u, (m_1m_3, Q-1) = 1, m_2|(Q-1)^v$，$u, v$是满足条件的最小自然数。

那么对于三个模数分别解决。由于$(m_3, Q) = (m_3, Q-1) = 1$，所以模$m_3$的答案可以用上述做法算出。

模m1时：由于$m_1|Q^u$，当$i \ge u$时，$P(i)Q^i \equiv 0 \pmod{Q^u}$，不需要计算。所以可以通过前几项直接算出。

模m2时：设$m_2 = p_1^{a_1}p_2^{a_2}\cdots p_k^{a_k}$，容易看出$v \le \max(a_1, a_2, \cdots, a_k)$。

根据题目条件M与$2, 3, \cdots, d+1$互质，所以M的最小素因子不会小于17。

因为$17^{14} > 10^{18}$，所以$v \le \max(a_i) \le 14$，即M与$2, 3, \cdots, d+v$互质。

$P(i)Q^i = P(i)((Q-1)+1)^i = P(i)\Sigma_{j=0}^{i}\binom{i}{j}(Q-1)^j$

当$j \ge v$时，$(Q-1)^j \equiv 0 \pmod{(Q-1)^v}$，可以忽略。所以

$\Sigma_{i=0}^{n-1}P(i)Q^i = \Sigma_{j=0}^{v-1}(Q-1)^j\Sigma_{i=0}^{n-1}P(i)\binom{i}{j}$

其中$\binom{i}{j}$是一个关于i的j次多项式。所以$\Sigma_{i=0}^{n-1}P(i)Q^i$是一个次数不超过$d+v$的多项式

令$G(n) = \Sigma_{i=0}^{n-1}P(i)Q^i$，可以暴力算出$G(0), G(1), \cdots, G(d+v)$，由于$M$与$2, 3, \cdots, d+v$互质，所以可以算出$G(n) \bmod m_2$的值。

时间复杂度：$O(D+\log n)$

空间复杂度：$O(D)$

4.3 NOV12 COUNTARI

Description

给出n个数a_i，求有多少个三元组(i,j,k)满足$i < j < k$且$a_i - a_j = a_j - a_k$。

$n \leq 10^5$，$a_i \leq 30000$

Analysis

对序列进行分块，然后分两种情况进行统计：

一、i,k至少有一个和j同块，那么只需要预处理出前缀/后缀每个数字的个数，然后枚举j和i/k即可。

二、i,k都不和j同块，那么枚举j后把两边FFT起来就可以求出两边各取一个数能组合出的那些数。

当然由于常数巨大，所以并不能比$O(n^2)$暴力更快。

时间复杂度：$O(n\sqrt{n\log n})$

空间复杂度：$O(n\sqrt{n\log n})$

4.4 NOV12 MARTARTS

Description

有一场比赛，你和对手分别有n个人，你知道某两个人对战会得到的比分$a_i : b_i$，你可以安排哪两个人进行对战，但是你的对手（在你决定之

后）会选择一场去掉。

你的目标是最大化$\Sigma a - \Sigma b$的同时最大化Σa，你的对手目标是最大化$\Sigma b - \Sigma a$的同时最大化Σb，对手的决策会根据他的目标来决定。

求你最多能拿到多大的比分。

$n \leq 100$

Analysis

一个暴力的做法是枚举每一条边，将它的边权置为无穷大，进行一次KM，然后判断剩下的匹配边是否都小于这条边更新答案。这样是$O(n^5)$的。

但是注意到相邻两次之间只有两条边边权不同，只需要对于这两个点重新匹配即可，其他点的匹配和点权都可以直接使用。可以优化掉一个n变成$O(n^4)$。

时间复杂度：$O(n^4)$

空间复杂度：$O(n^2)$

4.5 OCT12 MAXCIR

Description

给定平面上三个点A, B, C和n个向量，你可以选取至多K个向量，然后使A加上这些向量。你的目标是最大化$|AB| + |BC| + |AC|$的值。

$n \leq 500$

Analysis

由于$|BC|$固定，只需要最大化$|AB| + |AC|$即可。那么解的值相同的点形成了一个椭圆。求出最优点在椭圆上的切线，那么只需要把所有高于这条线的向量都选上，低的不选上就可以取到最大值了。

由于切线的斜率经过两个点之间的连线时会改变两个点的顺序，经过原点和某个点的连线时会改变这个点的正负性，所以只有$O(n^2)$种对点排序的策略，对于每一种顺序二分求解取最优值即可。

时间复杂度：$O(n^2 \log n)$

空间复杂度：$O(n^2)$

4.6 SEPT12 PARADE

Description

给定一张n个点的带权有向图，需要找出一些路径，路径经过的边会支付边权的费用（经过多次则支付多次），没有被任何一个路径访问到的点需要支付C的费用，起点和终点不相同的路径也需要支付C的费用。给出m组询问，每组询问给定C，求最小化总费用。

$n \leq 250$，$m \leq 10000$

Analysis

由于只有访问一个点才能使费用减少，所以假如有两条路径都从同一个点出发或者都在同一个点结束，那么可以使一条路径的长度缩短一格，访问的格点数不变但是代价减少。所以从每个点最多只会有一条路径出发和一条路径进入。

这种转化让我们想到了二分图匹配。以两个点之间的距离作为边权建出一个二分图，然后在上面增广。每一条增广路都代表减少一个C并增加一个增广路长度的费用。那么将询问排序并且每次找出增广路和C进行比较，假如更新会变优则更新。那么询问的时间就被均摊了，复杂度就是一个正常二分图匹配的复杂度。

时间复杂度：$O(n^4 + m \log m)$

空间复杂度：$O(n^2 + m)$

4.7 SEPT12 KNGHTMOV

Description

给定棋盘上一个点(m, n)，要从$(0, 0)$出发走到(m, n)，每一步可以走(x_1, y_1)或者(x_2, y_2)。有k个点不能经过。求走到终点的方案数或者输出无穷多或者输出无解。所有坐标均不超过M

$M \le 500$，$k \le 15$。

Analysis

首先根据两种走法是否线性相关分成两种情况进行讨论：

一、线性相关时：

由于走超过x_1步(x_2, y_2)以后显然会形成环，所以可以把两种走法都走500步能到达的点全部处理出来。这种点不超过$O(M^2)$个。接下来直接将这个图当成一般图进行处理即可。

一般图的做法非常简单，即首先BFS确定是否有解，然后使用拓扑排序确定是否有环，有环则无穷多解。最后在DAG上进行dp即可。

二、线性无关时：

由于这时所有可能到达的点的个数可能到达M^3个，所以不能直接建图判断。

这时可以使用容斥，即对于每个点（包括终点和障碍）算出起点到这个点的路径数（不计算障碍），然后使用容斥，即减去从每个障碍到达这个点的方案数，就可以算出起点不经过障碍到达终点的方案数。时间复杂度为$O(k^2)$。

时间复杂度：$O(k^2 + M^2)$

空间复杂度：$O(M^2)$

4.8 AUG12 GTHRONES

Description

纸上有n种数，每种数有很多个，两个人轮流行动，每次每个人需要取走一个数，除了第一次之外，这个数必须和上一次对方取的数“相邻”。无法操作的人输。求哪一边会胜利。如果先手胜利输出第一轮能取的最小数。

相邻的定义：对于两个不同的数i, j，满足$i|j$且$\frac{j}{i}$为质数或$j|i$且$\frac{i}{j}$为质数。

$n \le 500$

Analysis

首先可以使用miller-rabin暴力判断两个点是否相邻。由于一个点最多只和$O(\log n)$个比它小的点相邻，所以边数是$O(n \log n)$的。

那么容易发现这是一个二分图，而且后手必胜的条件就是这个二分图有完备匹配。检查完备匹配可以用网络流实现。

那么先手就需要在选走一个点之后使得剩下的图仍然有完备匹配。于是只需要删除满流边后从起点和终点各进行一次BFS，能BFS到的点就一定是可被选的。因为假如能BFS 到那么就可以用这条路径代替从起点到这个点的流量，于是这个图仍然具有完备匹配。

时间复杂度：$O(n^2 \log n)$

空间复杂度：$O(n \log n)$

4.9 AUG12 MAGIC

Description

给定一张n个点m条边的无向图，A在1号点，B在2号点，两个人各有P点法力值，两个人轮流行动，每一次行动可以依次：

- 沿着边走到一个点，假如走到了对方所在点那么胜利
- 加入一条当前没有的边
- 假如还有剩余法力值那么可以消耗一点法力传送到任意点

求谁胜利。

$n \leq 7777$

Analysis

容易发现每个块大小并不重要，只需要求出奇偶性即可。同理加上不影响连通性的边数也不重要，只需要奇偶性即可。

首先特判一些情况：假如两个人初始在同一块那么先手胜。假如n为奇数那么最后一定是两个人各占一块，而且边数固定。那么赢家也固定。

当n是偶数的时候，一定有一方想要使最后剩两块变成两个偶数块（不妨称为P），而另一方想要变成两个奇数块。

假如不能瞬移：那么当两个人的初始块都是奇数的时候P一定失败，因为对手始终都可以保持一块奇数。同理当两个人的初始块都是偶数的时候P一定获胜。当一奇一偶的时候就是先手获胜。

假如能瞬移：P是先手时，假如至少有一块是偶数，那么P一定可以在最后只剩3块的时候瞬移到偶数上就赢了。否则假如有至少4块奇数，P可以逼迫对手先使4块变成3块然后瞬移获胜。当P时后手时同理，只不过偶数块至少要两个否则先手可以破坏那个偶数块。

时间复杂度：$O(n)$

空间复杂度：$O(n)$

4.10 JULY12 EST

Description

给定一个字符串S，求所有和这个串的后缀树相等的串的数量。

$|S| \leq 10^5$

Analysis

容易发现满足条件的串必须是长度和S相等并且拥有的字母数和S相同。那么可以计算出最小表示后的串的数量然后乘上字母组合数。

观察S的后缀树，可以发现存在一个K，满足长度为$1 \sim K$的串都不存在对应深度的叶子，长度为$K+1 \sim |S|$的串都存在。那么也就是说只要确定长度为K的后缀在后缀树上的位置就可以确定所有后缀的位置了，也就是说相同的最小表示最多只有$|S|$种，可以暴力枚举然后判断。

首先考虑如何求出K：只需要把串反序然后对自己做一遍KMP取最长的即可。

求出了K以后就可以枚举长度为K的后缀和哪一个后缀的LCP是K，使用扩展KMP判定可行性，并使用Hash去重即可。

时间复杂度：$O(n)$

空间复杂度：$O(n)$

4.11 JULY12 DGCD

Description

给定一棵n个点的树，每个点上有权值。每次可以在一条链上加上一个数或者查询一条链上的数的gcd。

$n \leq 50000$

Analysis

首先进行树链剖分，就可以通过加一个log将树上问题简化成序列问题。那么接下来考虑序列问题的做法。

对序列进行差分，那么修改只需要修改$O(1)$个结点即可。由于差分后算值时还是需要用到第一个结点原来的值，所以还需要另外开一个线段树维护每个点的权值。

时间复杂度：$O(n \log^3 n)$

空间复杂度：$O(n)$

4.12 JUNE12 MATCH

Description

给出一个n+m个点的二分图，每条边出现的概率给定，问最大匹配的期望。

$n \leq 5, m \leq 100$。

Analysis

容易想到一种做法：将右边的点一个一个加进来并维护每种匹配情况的出现概率。但是光维护最大匹配是不够的，还需要实时维护所有可能的匹配的情况。虽然所有可能的匹配的情况看起来高达2^{2^n}种，但是暴力求出所有的状态之后发现其实只有406种。

那么每次先求出所有的状态，然后将右边的点按顺序加进来并枚举所有2^5种连边情况，然后在dp数组中转移即可。

时间复杂度：$O(406*m)$

空间复杂度：$O(406)$

4.13 JUNE12 COOLNUM

Description

有一个k位数，每个数位上的数字分别为$X_1, X_2, \cdots, X_K$，如果一个数n存在一到三个数位上的数的和为s，且$(\Sigma X_i - S)^S$是n的倍数，那么n是cool number。

定义$LC(n)$和$RC(n)$分别是小于等于n的最大的cool number和大于n的最小的cool number，多次询问给定n，求$LC(n)$和$RC(n)$。

$n \le 10^{1000}$

Analysis

首先显然所有非零位不超过3个的数都是cool number，这类数可以很方便地求出。

那么考虑剩下的cool number，设$k = \log_{10} n$，必须要满足$(9k-27)^{27} > 10^{k-1}$，那么k不超过77，不妨预处理出所有这类cool number然后在查询时二分。

为了求出这些数，可以枚举$\Sigma X_i - S$，然后枚举$(\Sigma X_i - S)^{27}$的不超过10^{77}的约数，然后用背包判断是否满足条件（这里可以使用压位优化到$\frac{1}{32}$）。经过常数优化后能在1.5s所有跑出所有32741个数。

时间复杂度：$O(\log n)$

空间复杂度：$O(n)$

4.14 MAY12 TICKETS

Description

有m个顾客，第i个顾客来吃饭时需要提供一份a_i菜或一份b_i菜才能满足顾客的要求。现在有n个菜各一份，求最多可以卖出去多少张饭票使得无论哪些顾客来都能满足要求。

$n \le 200$,$m \le 500$。

Analysis

首先转化题目，发现就是在一张图中寻找一个点数最少的子图使得边数超过点数。

经过观察，发现点数为边数+1的时候只有两种情况：

第一种：两个点之间连了三条路径。这个可以通过枚举两个点直接BFS算出。

第二种：两个环被一条链连了起来。首先枚举一个点当根，然后进行BFS建出一棵树。这样每个点到根的最短路就是它的深度。那么检查所有环，每个环的代价为环长+环到根的距离。取最小的两个环相加即可。

时间复杂度：$O(n^2m + n^3 \log n)$

空间复杂度：$O(m)$

4.15 MAY12 LEBOXES

Description

有n个盒子，每个盒子有a_i的几率装着b_i块钱，有$1 - a_i$的几率装着一块钻石。

有m个物品，每个物品需要c_i的钱和d_i的钻石来买。你每次会买尽量多的物品，求你能买的物品数量的期望。

$n \le 30$,$m \le 30$。

Analysis

首先需要计算出拥有几个钻石的时候购买几个物品最少需要多少钱。这个可以用一个三次方的dp求出。

接下来考虑使用搜索确定每种情况的概率，由于数据范围稍大，可以使用meet in middle解决。

首先将盒子分成两部分：前x个和后$n-x$个。对于前x个，暴力枚举所有2^x种组合方式，并按取得的钻石数分类，然后按钱数排序。

接下来暴力枚举后$n-x$个的组合方式，对于每种组合方式，枚举合并后最终拥有多少块钻石和购买多少物品，然后在预处理数组中二分，将这种情况出现的概率相乘计入答案即可。

时间复杂度：$O(n^3 * 2^x + n * 2^{n-x})$

空间复杂度：$O(nm + 2^x + 2^{n-x})$

4.16 APRIL12 CONNECT

Description

有一个$m \times n$的网格图，每个点有权值v_i和费用c_i。求一个最小的连通块使得至少出现K种正权值。

$m, n \leq 15$,$k \leq 7$,$v_i \leq m \times n$。

Analysis

假如每个点的权值不超过k，那么这就是一个经典的斯坦纳树问题，可以在$O(3^k nm)$的时间内解决。

考虑随机一种权值的映射，将每种权值映射成不超过k的值，这样假如最优解中的值被映射成了不同的值问题就解决了。那么只需要重复多次取最优解即可。

时间复杂度：$O(3^k mn)$

空间复杂度：$O(2^k mn)$

4.17 APRIL12 TSUBSTR

Description

给定一棵n个点的树，树的节点上有字母，将每个点到根的路径上的串的子串全部提取出来，第一问：求子串个数。第二问：m个询问，每次给出一个字母大小的顺序，求在这个顺序下第k小的串是什么，如果没有输出-1。保证输出文件大小不超过800K

$n \leq 250000, m \leq 50000$

Analysis

首先建立出这棵树的后缀自动机，那么第一问很好解决，答案就是$\Sigma length_i - length_{fail_i}$

对于第二问，首先需要预处理出沿着转移边走，到每个点之后有多少种子串。这个可以通过按照$length$降序dp解决。那么对于每个询问，从起点开始走，按照字母的增序判断走到这个点后剩余的子串数是否足够。由于保证了输出文件大小，所以这部分的复杂度是$O(Output\Sigma)$的。

时间复杂度：$O(n\Sigma)$

空间复杂度：$O(n\Sigma)$

4.18 MARCH12 EVILBOOK

Description

有n个人，打败第i个人需要a_i点代价，但是能获得b_i点魔法。可以使用m点魔法使一个人的a_i和b_i都变成原来的$\frac{1}{3}$。求使自己的魔法变成至少666需要付出多少代价。

$n \leq 10$,$10 \leq m \leq 666$

Analysis

由于一个人能获得的魔法假如比花费的魔法还少那么就没有意义。同时假如一个人的魔法超过666那么能让它变少就会变少。所以每个人的可行

状态非常少，直接搜索即可。

容易发现确定了所有人的削减次数以后一定是先从削减次数少的人开始打，所以直接按顺序搜索再加一些剪枝即可。

时间复杂度：$O(4^n)$

空间复杂度：$O(n)$

4.19 MARCH12 CIELQUAK

Description

给出一张$m \times n$的网格图，每条边有x的概率损坏，求$(1,1)$和(m,n)连通的概率。

$n \leq 8, m \leq 10^{18}, x \geq 0.1$

Analysis

当m较小时可以使用轮廓线DP，记录下轮廓线上和起点的连通状态并最小表示防止重复即可。当$n=8$时仅有不到4000个状态。

由于保证了x的大小，容易发现答案变小的速度非常快，也就是精度变得不那么重要了，那么可以求出$m = 1 \sim p$的值，设为$F(x)$，那么$F(m)$可以用$(\frac{F(p)}{F(p-1)})^{m-p} * F(p)$来近似。$p$取30左右即可满足精度要求。

时间复杂度：$O(n^n)$

空间复杂度：$O(n^n)$

4.20 FEB12 FINDSEQ

Description

给定n个数，要求从中选出5个数使得它们按顺序排列时大小关系是给定情况。

$n \leq 1000$。

Analysis

枚举第二个数和第四个数，那么剩下每个数可以按照从大到小（或从小到大）的顺序依次贪心选取区间内最大的可行的数。

那么就需要支持查询区间内小于某个数的最大数。维护二维前缀和后二分即可。

时间复杂度：$O(n^2 \log n)$

空间复杂度：$O(n^2)$

4.21 FEB12 FLYDIST

Description

给定一个n个点m条边的整数边权无向图，你需要使每条边的边权变成两个点之间的最短路长度，而且每条边改变之后必须是非负，设边权原来为O_i，改变之后为N_i，代价为$\Sigma|O_i - N_i|$，求最小化代价，结果以分数形式输出。

$n \leq 10, O_i \leq 20$

Analysis

考虑使用线性规划解决本题，首先将每对点之间的最短路当成变量，设为$d_{i,j}$，并对每条边设两个变量I_k, D_k，表示最终边权为$O_k + I_k - D_k$。

对于每条边k，设连接的点为x_k, y_k，那么对于每个j都有$d_{x_k,j} \leq O_k + I_k - D_k + d_{y_k,j}$，同理反向也有对应的一个不等式。同时为了保证每条边边权非负需要加上$O_k + I_k - D_k > 0$，使用单纯形解出这个线性规划即可。

时间复杂度：$O(n^2)$个变量$O(n^2)$个式子的线性规划复杂度

空间复杂度：$O(n^2)$

4.22 JAN12 CARDSHUF

Description

给定一个大小为n的栈，初始时里面元素为1到n。有m个操作，每次操

作给定三个数a, b, c，你需要：

- 取出最上面的a个数
- 再取出最上面的b个数
- 把之前的a个数放回去
- 取出最上面的c个数
- 把之前的b个数翻转顺序以后放回去
- 把之前的c个数放回去

求最后栈的形态。

$m, n \leq 100000$

Analysis

本题其实只需要支持三个操作“取出一段序列”,“翻转一段序列”,“连接两段序列”。只需要用一棵平衡树维护这个序列然后模拟操作即可。

时间复杂度：$O(n \log n)$

空间复杂度：$O(n)$

4.23 JAN12 MISINT2

Description

定义一个字符串是“好的”当且仅当将这个字符串的所有偶数位按顺序取出放到串的最前面之后和原串相等。

给定l, r，求长度区间$[l, r]$内有多少“好的”字符串。

$l, r \leq 10^{10}, r - l \leq 50000$

Analysis

设$g(n)$为长度为n时的答案，$f(n)$为长度为n时的置换的循环数，那么就有$g(n)=26^{f(n)}$，求出$f(n)$即可。

由于长度为偶数的串最后一位单独成环，就是比他小1的奇数的答案+1，于是只需要计算所有n为偶数的答案即可。

设ord_i表示2对i的阶，经过观察可以发现$f(n)=\Sigma_{p|(n+1)}\frac{\phi_p}{ord_p}$。由于$n$为偶数，于是$p$为奇数，也就是说2对$p$的阶一定存在。

首先需要求出所有合法的p。直接pollard-rho的复杂度是$O((r-l)r^{0.25})$的，无法承受。而使用筛法枚举小于$\sqrt{r}$的约数进行筛选的复杂度是$O(\sqrt{r}\log r)$的，可以接受。

由于ord_i和ϕ_i都是积性函数，在DFS找p的时候可以顺便维护ord_i和ϕ_i，就可以$O(1)$计算了。但是注意到每个数会有至多一个比较大的质因子，需要每次暴力计算ord_i。由于这部分并不是耗时的重点，$O(\sqrt{n})$暴力计算ord_i即可。

时间复杂度：$O(\sqrt{r}\log r+(r-l)\sqrt{r})$

空间复杂度：$O(\sqrt{r}+(r-l)\log r)$

5 传统题2011

5.1 DEC11 SHORT2

Description

给定质数p，问有多少对$a,b(a>p,b>p)$满足ab被$(a-p)(b-p)$整除。

$p\leq 10^{12}$

Analysis

原题等价于求$ab|p(a+b+p)$的对数。那么分三种情况讨论：

一、a,b都不被p整除：首先容易发现，除了$a=b=1$，否则$a\neq b$。不妨假设$a<b$，那么$b=\frac{a+p}{ka-1}$，所以a和$ka-1$的值的范围是$O(\sqrt{p})$的。那么枚举$ka-1$和b中较小的一个统计答案即可。

二、a,b都被p整除，答案只有5种直接加上去即可。

三、a,b有一个被p整除，容易发现（一）中的每种情况分别对应这里的两种情况：(a,b)变为$(a,\frac{p(a+p)}{b})$和$(\frac{p(b+p)}{a},b)$，直接将答案*3即可。

时间复杂度：$O(\sqrt{p})$

空间复杂度：$O(1)$

5.2 DEC11 HYPER

Description

定义3-超图是一个类似普通图的图,只不过其中的边都连接三个点。

定义3-超树是一个去掉任意一条边都以后都不连通的3-超图。

给定n,问有几种含有n个带标号的点的本质不同的3-超树。

$n \leq 17$

Analysis

首先特判$n=3$的情况，接下来只考虑$n \geq 3$的情况。

定义点双连通分量为去掉任意一个点依然连通的图。那么首先把一棵超树它分成若干个点双连通分量，可以发现对于一个点双连通分量，每一条边连接的三个点中都恰好有一个点为叶子节点。把那个节点删掉就得到了一个正常的的点双连通图。

那么暴搜出点数为i的双连通超树数量，然后暴力枚举要求的超树的每一个点双连通分量的大小，然后使用记忆化搜索求出这些点双连通的超树拼接而成的不同的超树数量。

直接提交并不能在时限内出解，但是注意到n很小，打表提交即可。

时间复杂度：$O(1)$

空间复杂度：$O(1)$

5.3 NOV11 LUCKYDAY

Description

给定一个递推数列S满足$S_1 = A, S_2 = B, S_i = (X * S_{i-1} + Y * S_{i-2} + Z) \bmod P$。

接下来有m组询问，每次给出l, r，求$l \le k \le r$且$S_k = C$的k的个数（C为定值）。

$m \le 20000, P \le 10007, P \in prime$

Analysis

当X和Y中至少有一个为0时，循环节的长度不超过P，暴力求解即可。接下来只考虑X和Y都不为0时的情况。

首先把每个数S_i表示成$pair(S_{i-1}, S_i)$，这样只需要枚举到相同的pair就可以确定循环节了。

由于循环节长度可能达到P^2，所以不能暴力枚举。但是注意到符合条件的数只有P个（第二位确定第一位为任意数），可以考虑分块。即每隔$\sqrt{P}$个数求一次值，将求到的$P\sqrt{P}$个数存进hash表中，然后枚举符合条件的数进行判断。判断的时候直接枚举模$\sqrt{P}$的位置判断即可。

求出了一个循环节内所有符合条件的位置之后（最多P个）只需要每次询问时二分一下即可。

时间复杂度：$O(P\sqrt{P} + m \log P)$

空间复杂度：$O(P\sqrt{P})$

5.4 OCT11 PARSIN

Description

给出M, N, X，求$\Sigma_{k_1+k_2+\cdots+k_M=N} \prod sin(k_i X)$。

$M \le 30, N \le 10^9$

Analysis

定义$f_{N,M} = \Sigma_{k_1+k_2+\cdots+k_M=N} \prod sin(k_i X)$

$g_{N,M} = \Sigma_{k_1+k_2+\cdots+k_M=N} \cot(k_M X) \prod sin(k_i X)$

那么$f_{i,j} = (f_{i-1,j-1} + g_{i-1,j} sin(X) + f_{i-1,j} cos(X)$

$g_{i,j} = (f_{i-1,j-1} + g_{i-1,j} cos(X) - f_{i-1,j} sin(X)$

使用矩阵乘法优化转移即可。

时间复杂度：$O(m^3 \log n)$

空间复杂度：$O(m^2)$

5.5 OCT11 BAKE

Description

有一个7维空间，每一维都为10左右。有n个操作，每次操作可以加入一个带权点，每个点的有些维度（除了最后一维）可以是任意，代表查询任意一个位置都将它算在内。每次操作还可以查询前六维固定（或者为任意），第七维在一个区间内的点的权值和。

$n \leq 10^5$

Analysis

用一个七位数组存下每个位置的贡献，加点查询直接暴力。

加入带有任意的点时直接将所有会影响到的位置暴力加上贡献即可。

时间复杂度：$O(n)$

空间复杂度：$O(1)$

5.6 SEPT11 SHORT

Description

给定n, m，求满足$(a-n)(b-n)|ab-n, n < a < k, n < b < k$的$(a,b)$的对数。

$n \le 10^5, m \le 10^{18}$

Analysis

首先特判$n = 0$，那么当$n \neq 0$时答案都比较小，可以暴力求出所有答案。

不妨设$a \le b$，那么$a \le 2n + \sqrt{2n^2 - n}$，所以暴力枚举$a$是$O(n)$的。

由于$b = n + \frac{n^2+an}{ap-an-a}$，那么直接枚举$p$或者枚举$n^2 + an$的约数都会超时，需要将两种做法结合起来，首先打表找出两种做法时间差不多的时候mid，然后对小数据分解约数大数据枚举p就能通过了。

时间复杂度：$O(n\sqrt{N})$

空间复杂度：$O(n)$

5.7 SEPT11 CNTHEX

Description

求满足以下条件的六边形个数：

- 木根长度都不超过n
- 每种长度的木棍都不能超过k
- 最长的木棍长度不小于l，其他的木棍长度不大于x

答案模$10^9 + 7$。

$1 \le k \le 5, 1 \le x \le l \le n \le 10^9, n - l \le 100$。

Analysis

令$f_{w,p,mask,i,j}$表示现在dp到第w位的第p个数，当前相邻两数是否相等的二进制状态为$mask$，当前所有小的数加起来比最大数小i，第$p-1$个数的这一位是j使用数位dp解决即可。转移的时候需要记下前一个w的前缀和。

注意到这样可能会计算进长度为0的木棍，于是将所有木棍+1最后减掉即可。

时间复杂度：$O((n-l)2^k k^2 \log n)$

空间复杂度：$O(2^k + \log n)$

5.8 AUG11 SHORTCIR

Description

给定一个由变量,and,or,not组成的表达式，每个变量的值只能为0或1，给出每个变量为1的概率，为了计算出表达式的值需要将变量以某种顺序计算（由于一连串的and或or计算出一个不符合后剩下的可以不用计算），使得期望计算次数最少。

表达式长度$|S| \le 30000$。

Analysis

首先可以根据表达式建出一棵树，对于每个结点记录下为1的概率p_i和期望询问次数w_i。由于and和or优先级不同，所以一个节点的儿子只能全是or或全是and。

使用排序不等式可以很容易地看出or时按照$\frac{w_i}{p_i}$递增的顺序，and时按照$\frac{w_i}{1-p_i}$递增的顺序最优。那么对于每个结点的儿子进行排序算出这个结点的值即可。

时间复杂度：$O(|S| \log |S|)$

空间复杂度：$O(|S|)$

5.9 AUG11 DIVISORS

Description

给定正整数B和X，求满足至少存在一个数$D(N < D \le B)$能整除$N * X$的正整数N的个数。

$B \le 10^{12}, X \le 60$

Analysis

考虑枚举$i=\frac{NX}{D}$来统计。为了防止重复，i必须是满足条件中最大的那个数。

设枚举到K时的答案是$f(K)$，那么

$$f(K)=\Sigma_{S\subseteq\{K+1,\cdots,X-1\}}\frac{(-1)^{|S|}B\gcd(K,X)}{Xlcm(\frac{x}{\gcd(x,lcm(X,K))})for\ x\in S}$$

直接计算复杂度无法接受，但是注意到LCM的可能值非常少，只有10000个左右，可以全部处理出来然后使用容斥计算。

时间复杂度：$O(10^4X^2)$

空间复杂度：$O(10^4X)$

5.10 JULY11 YALOP

Description

有一个$m\times n$的01矩阵，有一个人要从$(1,1)$走到(m,n)，每次可以走向周围的八个格子之一。每当他走出一个格子，这个格子和他周围的四个格子的值就会取反。问是否能够使得每个格子都变成0。

$\max(m,n)\le 10^9$,$\min(m,n)\le 40$,1的个数≤ 10000

Analysis

注意到在一个2×2的方格内走$(1,1)\to(2,2)\to(1,1)\to(2,2)\to(1,2)$就可以走到相邻的格子而不改变棋盘状态。也就是说，当$n,m\ge 2$时，任意一种翻转格子的方案都是可以做到的。

不妨设$2\le m\le n$。由于每个格子翻转多次是没有用的，所以只可能翻转一次或零次。容易发现，确定第一行是否翻转以后，剩下所有的状态都可以确定了。于是就得到了一种暴力做法：枚举第一行翻转方法，求出状态然后判断是否可行。那么考虑如何加速这个算法。

首先考虑将一个值为1的格子通过一系列翻转使得只有第一列有值为1的格子。这个显然可以$O(n)$做到。但是当n大的时候就不能直接暴力

算了。由于m很小，于是它的循环节也很小，所以打出循环的表以后就可以$O(1)$求出每个格子对第一列的影响了。

读入的时候直接处理出影响完第一列以后的第一列的状态。那么现在要求的就是找出一种选择最后一排翻不翻的方案，使得第一列恢复全零的状态。这个可以通过高斯消元求出。到此为止就解决了$m \geq 2$的情况。

当$m = 1$时，由于不是每种状态都能达到了，所以多了一个不翻转的格子个数必须为奇数的限制，特判即可。

时间复杂度：$O(m^3)$

空间复杂度：$O(m^2)$

5.11 JULY11 BB

Description

求有多少个长度为n的01串满足任意连续的m个数字中至少有k个1，同时1的个数最少。答案模$10^9 + 7$。

$k \leq m \leq 50$,$n \leq 10^9$。

Analysis

首先考虑$m|n$的情况，那么最少的1个数显然是$n/m * k$。考虑将串按m个一组划分，那么每一组显然必须是k个1，然后将每一组的1的位置写在$m * (n/m)$的矩阵内。比如000111 001101 100110 110100写成

1 2 4

1 4 5

3 4 6

4 5 6

可以看出这个矩阵每行内元素递增，每列内元素不降。容易证明这个结论和原题条件是等价的。

那么让第i列的元素都减去i，那么每行内元素也变成了不降。那么就变成了求矩阵的个数，其中每个元素在0到$m - k$之间。其中行数和数字范围都很小，列数比较大。而这个矩阵的计数可以转变成标准杨氏矩阵的计

数。详见这里。

考虑m不整除n的情况。假如余数较小，那么最后一段就不包含任何一个1，可以直接计算。假如余数较大，那么最后一段就会包含$k-(m-n\%m)$的1，并且最后几位去掉的都是1。这种情况可以转变成另一个行列数不同的矩阵求值，可以直接使用之前的过程求出。

时间复杂度：$O(m^2)$

空间复杂度：$O(1)$

5.12 JUNE11 CLONES

Description

定义一个函数$f: A \to B$为布尔函数，其中A为长度为n的01串，B为0或1。

给定4个由布尔函数组成的集合Z, P, D, A，其中

- Z是所有满足$f(0, 0, \cdots, 0) = 0$的集合。
- P是所有满足$f(1, 1, \cdots, 1) = 1$的集合。
- D是所有满足$!f(x_1, x_2, \cdots, x_n) = f(!x_1, !x_2, \cdots, !x_n)$的集合。
- A是所有满足如果$f(x_1, \cdots, c, \cdots, x_n) = f(x_1, \cdots, d, \cdots, x_n)$则$f(y_1, \cdots, c, \cdots, y_n) = f(y_1, \cdots, d, \cdots, y_n)$的函数，在这里$c$和$d$都在同一个位置$i$，并且对于任意$i, x_1 \sim x_n, y_1 \sim y_n, c, d$都成立。

给出m个由$Z, P, D, A, \backslash, \wedge, \vee, (,), !$组成的表达式，其中$\backslash, \wedge, \vee, !$表示集合的差、交、并、补集。求出每个表达式代表的集合的大小。表达式长度不超过L

$n, m, L \leq 100$。

Analysis

根据每个函数是否属于Z, P, D, A，将它们用二进制数表示。容易看出使用同一个二进制数表示的函数是等价的。那么首先求出每个二进制数代表多少个函数。这个可以通过推公式或者打表得到。

那么一个集合只可能完全包含或者完全不包含某一个二进制数。那么就可以将一个集合用一个16位的二进制数表示。集合运算变成了整数运算。直接算出表达式的值并带入每个二进制数的答案即可。

时间复杂度：$O(m(L+\log n))$

空间复杂度：$O(L)$

5.13 JUNE11 MINESREV

Description

给定一个$m \times n$的扫雷游戏，每次操作可以打开一个方块，然后所有能使这个方块打开的方块（即那些周围没有雷并且和和这个方块相连的方块）都会被自动打开，问最少需要几次点击才能打开所有方块。

$m, n \leq 50$

Analysis

将那些周围没有雷的方块称为空白方块，那么除了空白方块剩下的方块都需要一次点击才能打开，只需要考虑空白方块即可。

空白方块可以分为多个连通块，每个连通块内都是同时开启或关闭。并且假如有一个数字方块与两个空白连通块相邻就可以通过一次点击同时打开两个块（显然最多只能同时和两个相邻）。

以空白连通块为点建出一张图，那么每次可以同时打开两个相邻的点，那么答案就是总点数减去最大匹配数，可以使用带花树解决。

时间复杂度：$O(m^2n^2)$

空间复杂度：$O(mn)$

6 challenge题

6.1 FEB15 CHPUZZLE

Description

给定一个$m \times n$的网格图和k块拼图，每块拼图都是一个连通块，拼图的位置可以平移，但是不能旋转。

现在你需要放置这些拼图，使得方格图被覆盖的面积尽量大。拼图之间不能互相覆盖，拼图不能超出方格图范围

$m, n \leq 1000$

Analysis

由于直接计算拼图形状比较麻烦，考虑使用外接矩形来代表每一个方块。

假如将这些矩形按照某个顺序排序之后，那么每个矩形放置的位置一定和之前放置的矩形相邻。

那么每次以(x, y)为左上角放置一个(a, b)的矩形，就将$(x, y+b)$和$(x+a, y)$置为新的可用点，每次放置新的矩形就从可用点中通过估价选出一个即可。

6.2 JAN15 SEAND2

Description

给定一个整数A，在它的十进制表示中不含零。再给出n个整数B_i。

定义关于数字A的一个函数f：$f(A) = \Sigma_{i=1}^{n} A_i \pmod{B_i}$（即$A$的数位和）

现在你需要对A中的数字进行重排，来最小化$f(A)$的值。

$A \leq 10^{1000}, n = 100, B_i \leq 10^6$

Analysis

可以使用调整法，即每次交换两个位，假如比现在的解优就做否则就不动。那么问题的关键在于如何快速的求出交换两个位后的答案。

首先可以break掉两个数相同的情况，可以节省10%的时间。对于每个模数，每次交换只需要重新计算两个位置对总和贡献的改变量就可以了。那么在预处理出10^i之后就可以$O(n)$计算是否变优了。

由于题目是多组数据的，可以首先先对所有数据做一遍，然后对于那些解不是非常优的多做几遍来提高分数。

6.3 DEC14 KALKI

Description

给定二维平面内的n个点，要求返回这些点的一个生成树，使得c_i的最大值最小。

c_i的定义：对于每个点，设在生成树中与其相邻的点钟最远的点的距离为R，那么以该点为圆心，半径R以内的的点的C_i全部增加1（包括自身）。

$n \leq 400$

Analysis

容易发现一个点的C_i至少是这个点的度数+1，那么就需要尽量降低每个点的度数。

这里我使用了两个点度数+1的8次方的乘积来做估价，直接按照估价每次选取当前估价最小的那条边求出最小生成树就可以获得不错的分数。

由于边权是动态的，直接求最小生成树会TLE，所以可以只选用最短的$\frac{1}{15}$左右的边来加速计算过程。

本地gen出数据以后容易发现答案其实都非常小，那么一旦一个点随机出了不超过5的答案就可以break，这样就有更多的时间留给那些不是很优的数据。

6.4 OCT14 CHEFPNT

Description

给定一个$m \times n$的棋盘，初始时棋盘上有一些黑色的格子，其它的格子都是白色的。

每一步可以选择任意一个白色的格子和一个方向（左-右或上-下），如果选择左-右，那么就从选定的白色格子开始向左边依次把格子涂成红色，直到到达了棋盘边界或者到达了某个之前已经被染成了黑色或红色的格子时停止涂色；然后再从选定的格子开始，向右边进行同样的涂色操作。上-下与左-右类似。

游戏的目标是用尽可能少的步数把所有原来的白色的格子都涂成红色。

$m, n \leq 100$

Analysis

假设一开始全部是上-下涂，那么要做的就是将一些格子变成左-右涂，每改一条带来的次数贡献就是：纵向被切断的条数减去纵向长度本来就是1的个数再减1。那么假如每次贪心修改贡献最大的那个就能得到比较优的答案了。

直接判断显然是会TLE的，但是注意到贡献其实等于改变的这一条的长度减去上下两排的上-下涂个数再减1，那么对于每一行预处理出这个信息以后这个问题就变成了最大区间和问题，可以$m \times n$的时间找到最优解。

那么涂完色以后使用拓扑排序求出每条线涂的顺序输出即可。由于之前的做法不保证不出现环，所以每次涂色完成之后都要进行一次拓扑排序，在出现环的时候要将这种改法特判掉。

但是这样还是不够优，还是有些情况不能解决：比如说$2 \times n$的情况应该横着涂两条，但是这种情况就会把他忽略。于是在改一条的基础上再加上同时改宽度相同的2,3,4条就可以使得分变优许多（由于判断多条会让常数变得很大而且条数多了效果也就没那么好了所以加到4就差不多了）。

6.5 JULY14 GERALD09

Description

给定一个$m \times n$的空矩阵，要求在里面填上G,C,A,T以构成一个尽可能稳定的基因矩阵。基因矩阵的稳定性与其互不相同的子矩阵的数目相关——越接近K越好。

要求构建尽可能稳定的基因矩阵，即对于一组测试数据，设返回的答案中有W个互不相同的子矩阵，这组数据的得分将是$\frac{|W-K|}{K}$。

$m, n \leq 15$

Analysis

首先考虑只使用两种字母能拼出的不同矩阵个数的所有方案，那么假如要求的个数小于只用两种字母能拼出的最大的数就可以只用两种字母达到要求。

由于m, n非常小，可以打表打出所有情况下只用两种字母的随机矩阵的最大值，这样就可以通过表中的数据直接求出一个与答案非常接近的矩阵。假如超出了两种字母能达到的上限就使用全随机的四种字母的矩阵。

有了一个与答案非常接近的矩阵以后就可以使用随机调整法每次改变一个格子来逼近答案了，这样做需要快速求出一个矩阵的不同子矩阵数量。

有一个$O(n^4)$的做法可以解决这个问题。首先只枚举$1 * j$大小的矩形，去重后编号，这一步只需要扫一遍就可以了。然后每个$i * j$矩形都可以被分割成i个$1 * j$的矩形，那么使用类似之前的做法扫一遍就可以为每个矩形编号了。

6.6 MAY14 ANUMFS

Description

给定一个由一个矩形删掉几个边界上的小矩形后形成的图形，要求找出其中的一个黑匣子或者输出不在区域内。

有n架飞机，每架飞机有两个参数，表示搜寻范围R和费用C。每次可

以往一个地点(x, y)派出一架飞机，花费为$2 * (x + y) * C$，可以检测到所有满足$|a - x| + |b - y| \le R$的点(a, b)。

保证总存在$R = 0$的飞机。

$n \le 5000$，地图大小$\le 2 * 10^5$。

Analysis

假设已经求出了向每个位置派出每架飞机的花费C_i，能检测P_i的范围，那么不妨假设每次都能询问和总面积相同比例的大小，那么$C_i / \log(\frac{P_i^2 + (area - P_i)^2}{area})$最小的就是询问次数最少的。

但是由于搜寻范围非常的大，根本不可能对每个位置都估价，那么可以随机几个位置对每架飞机都估价，就可以获得不错的分数。

6.7 JUNE13 CHAORNOT

Description

给定n个数，要求从中选出尽量多的数使得不存在三个数构成等差数列。

$n \le 10^5$

Analysis

由于能选出的数比较少，只有1000个左右，所以假如每次假如一个数都把所有不能选的数标记出来，也只有$5 * 10^6$左右种，可以承受。

那么可以按照一定的顺序加入数字直到不能加入为止。可以对数组进行多次排序找出较优的那种。

然后再尝试删掉一个数的同时能否加入多个数来进行调整，就能获得不错的分数。

6.8 DEC12 WORDNINJ

Description

你正在玩一个拼单词游戏，每一轮，你会获得一些方块，你可以从这些方块中选择零个或一个获取，然后丢弃其他的方块。之后，你可以将你手上的字母方块拼成一个单词，然后获得一定的分数（当然也可以不拼）。当所有轮数结束以后游戏结束。

游戏中除了字母方块以外还有几种特殊方块：双倍单词得分方块，三倍单词得分方块，效果是将下一个单词的得分加倍；双倍字母得分方块，三倍字母得分方块，效果是将下一个单词的某个字母的得分加倍；空白方块，效果是可以当成任意一个字母使用。

单词的得分为所有字母的得分之和再加上特殊方块的加分再加上长度加分（长度为4,5,6,7分别加3,6,9,42分）。每个字母的得分不同（根据稀有度排列为$1\sim10$不等），空白方块在组成单词时不计分，但是在取得空白方块时可以获得分数。

拼完单词以后，这个单词的最后一个字母会留下，其他字母会消失，然后获得分数并继续游戏。

你的任务是最大化得分。

所有轮中给出的方块都会事先告知，可以拼出的单词表也是事先告知（所有单词长度都在$1\sim7$之间），所以可以使用离线算法。

总方块数$=10000$

Analysis

由于长度为7的单词有非常大的额外加分，可以尝试只拼长度为7的单词。

可以使用一个贪心算法来取方块：首先看到加分方块就取，否则根据选取该字母以后还可以拼出的单词数和该字母的得分估价，选取一个字母。假如所有字母选了以后都没有可拼出的单词就不选。

但是这样做并没有利用题目的离线性质，而是当成了在线来做。而其实在选取这个字母的时候就可以知道接下来会出现什么字母，所以会选择马上能组出来的而不是选择选项更多的。将上一个方法离线，加分方块还

是能取则取，但是字母方块先不决定取不取，而是攒齐7块以后再使用这些块拼一个单词。当然直接枚举会超时，所以需要一些优化和卡时。

也可以使用动态规划来解决这个问题。$f(i,j)$表示第i块剩下的字母是j的最优解。这样在转移的时候就可以和剩下的字母一起转移，能处理更优的情况。不过代价是复杂度也会变得非常高，需要更优秀的估价函数，进行优化和卡时才能获得比较高的分数。

6.9 SPET12 SIMNIM

Description

给定n个数，它们的异或和为0。要求将他们分成k组，每组中的数异或值都为0。需要最大化k的值。

$n \leq 1000$

Analysis

考虑一个一个地加入数字，每次存在一组数字和为0时就取出，看起来这样做的答案已经比较优秀了，再加上多次随机加入顺序就能得到不错的解。那么考虑如何实现这个想法。

由于数字最多只有64位，所以消元数组中最多只有这么多的数，那么每次加入一个新数字就进行一次消元，假如消成了0就把那一组异或为0的数取出，然后对于剩下的点暴力重构一个消元即可。

6.10 APRIL12 SIMGRAPH

Description

给定两张n个点的无向图，要求对第二张图重新标号使得两张图中重合的边尽量多。

$n \leq 75$

Analysis

首先随机出一个较优的初始解，这里每次判断需要$O(n^2)$的时间，所以只能做几千次。

使用模拟退火，每次交换两个点然后根据答案变化确定是否真的交换。由于只变化两个点，所以每次判断只需要$O(n)$的时间，就能做几十万次了。

进行参数调整之后就能通过本题。