杂题选讲

仓鼠

写在前面

- 讲课人水平不大行，也没做过啥题
- 所以看似杂题选讲，实则原题选讲/简单题选讲
- 一些题可能需要一点点前置技能
- 所以略微归纳了一下，讲部分题的时候会先讲一下前置技能

Bitwise Xor

给定一个长度为 n 的整数序列 a 和一个整数 K，求 a 有多少子序列，两两异或值至少是 K.

$n \le 3 \cdot 10^5, a_i, K < 2^{60}$

Bitwise Xor

- 若干个数两两异或的最小值必在排序后相邻两个取到
- 排序后直接用trie优化dp即可

MOD Problem

- 有T组询问，每组询问都是：给定n，请你求出$\sum_{i=1}^{n} n \bmod i$
- $0 \le T \le 1000000 \quad 1 \le n \le 10000000$

MOD Problem

- $\sum_{i=1}^{n} n \bmod i = \sum_{i=1}^{n} n - \left\lfloor \frac{n}{i} \right\rfloor \times i = n^2 - \sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor \times i$
- $\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor \times i = \sum_{i=1}^{n} d_1(i)$
- $d_k(i)$表示i的所有因子的k次方之和
- $d_1(i)$可以很方便地用线性筛求出，预处理其前缀和
- 时间复杂度$O(n + T)$

平方串

我们定义一个串为平方串当且仅当这个串非空而且它可以由两个相同串连接而成。例如 naivenaive 和 aaaaaa 为平方串，而 naiveevian 和 aaaaa 不是。

现在 zzq 拿到了一个很长的数字串（每个字符为 $0 \sim 9$），他想要随机地取出一个非空子串，如果这个子串没有前导零且为平方串，那么它的价值就为它的数值，否则它的价值为 0。

zzq 想知道他取出子串价值的期望，$\bmod 666623333$ 输出。

$n \leq 5 \times 10^5$

平方串

- 首先有个很经典的套路，叫做插点法
- 具体而言，枚举平方串的长度$2 \times \mathrm{len}$，然后把原串每len个插一个关键点，那么平方串最多跨越两个相邻关键点
- 考虑枚举这两个相邻的关键点是什么，求出最长公共前缀后缀，可以发现平方串是一个区间
- 这整个过程是$\sum_{k=1}^{N} \frac{N}{k} = O(N \ln N)$的

平方串

- 注意这时候，我们相当于用$O(N \ln N)$个形如起点在$[l, r]$中长度为p的串这样的描述，覆盖了所有平方串（的前半部分）
- 考虑处理一个前缀$F[i] = 10 \times F[i-1] + S[i]$
- 那么$[i, j]$构成的数就是$F[j] - F[i-1] \times 10^{r-l+1}$
- 注意需要处理前导零，因为平方串的性质，所以当$S[i+1] = 0$的时候可以直接把$F[i]$看成0

隔板法

- 有N个元素排成一排，要把它们分成K组，每一组必须是连续的，计数方案数
- 考虑N个元素有$N-1$个空隙，分成K组就是在$N-1$个空隙中选择$K-1$个空隙插上板子。这样组与组之间就被板子分隔开来了
- 方案数考虑用组合数，显然是$\binom{N-1}{K-1}$

不定方程解数计数

- 不定方程计数是隔板法的直接应用
- 有不定方程$X_1 + X_2 + \cdots + X_n = S$，每个量都是整数，对于每个$X_i$有限制$X_i \geq R_i$，计数这样的不定方程的解数
- 考虑令$Y_i = X_i - R_i + 1$，那么有$Y_i \geq 1$，方程变为$Y_1 + Y_2 + \cdots + Y_n = S + (1 - R_1) + (1 - R_2) + \cdots + (1 - R_n)$。这样的方程可以直接考虑其组合意义用插板法计数
- 那么所谓可以重复选择元素的组合数问题也就直接解决了

不定方程解数计数

- 有一些简单的应用
- 比如计数$1 \le X_1 \le X_2 \le \cdots \le X_n = M$的方案数
- 做法是把相邻两项差分，对差分变量做不定方程解数计数
- 例题：2019全国高中数学联赛二试第二题

不定方程解数计数

- 再来考虑这样一个不定不等式的计数问题
- 条件不变，方程变为不等式$A \le X_1 + X_2 + \cdots + X_n \le B$
- 用前面提到的转化，变成求$X_1 + X_2 + \cdots + X_n \le A$的方案数。这个可以写成一个不定方程$X_1 + X_2 + \cdots + X_n + X_{n+1} = A$，这里$X_{n+1}$是一个辅助变量、限制是非负。然后就变成了不定方程的问题
- 直接从插板法的角度来思考这个问题也可以得到这个结果。来看一个小例题：求$\sum_{i=0}^{n}\binom{i}{k}$

不定方程解数计数

- $X_1 + X_2 + \cdots + X_n = S$，对于每个$X_i$有限制$X_i \geq 1$。计数每种方案$\prod_{i=1}^{n} X_i$的和
- 反过去考虑插板法的组合意义，相当于插板完成后在每一组都任意选择一个代表球的方案数。那么列方程的时候变成两个变量，表示选的代表球之前的数目和之后的数目
- $(X_1 + Y_1) + (X_2 + Y_2) + \cdots + (X_n + Y_n) = S - n$，每个变量非负
- 那么算什么$\prod_{i=1}^{n} \binom{X_i}{K_i}$的和也是同样的做法

概率

- 基本事件空间Ω的元素ω_i称为基本事件，Ω的子集A被称为事件
- 如果给每个ω_i分配一个权重$P(\omega_i)$满足$0 \le P(\omega_i) \le 1$且$\sum_{\omega_i \in \Omega} P(\omega_i) = 1$，那么可以定义事件A的概率为$\sum_{\omega_i \in A} P(\omega_i)$
- $P(\emptyset) = 0$
- $P(\Omega) = 1$
- $A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)$
- 对于事件A，定义$\overline{A} = \Omega - A$，则$P(A) + P(\overline{A}) = P(\Omega) = 1$
- 称A和B两个事件是独立的，当且仅当$P(A)P(B) = P(A \cap B)$

概率

- $P(A|B)$表示在已知B发生的情况下发生A的概率
- $P(AB)$表示A和B同时发生的概率
- 全概率公式：假设$B_1, B_2, \ldots, B_k$是Ω的分割，那么$P(A) = \sum_{i=1}^{k} P(A|B_i)\,P(B_i)$
- 条件概率：$P(A|B) = \frac{P(AB)}{P(B)}$
- 贝叶斯公式：$P(A|B) = \frac{P(AB)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$

随机变量和期望

- 随机变量X是一个定义在Ω上的函数
- 随机变量X的期望定义为$\mathrm{E}[\mathrm{X}] = \sum_{\omega_i \in \Omega} \mathrm{X}(\omega_i)\mathrm{P}(\omega_i) = \sum_x x\mathrm{P}(\mathrm{X} = x)$
- 期望的线性性：$\mathrm{E}[\alpha \mathrm{X}_1 + \beta \mathrm{X}_2] = \alpha \mathrm{E}[\mathrm{X}_1] + \beta \mathrm{E}[\mathrm{X}_2]$
- 期望的线性性是一个很好的性质，它对任意两个变量X_1和X_2均成立
- 如果X_1和X_2独立，$\mathrm{E}[\mathrm{X}_1\mathrm{X}_2] = \mathrm{E}[\mathrm{X}_1]\mathrm{E}[\mathrm{X}_2]$
- 类似全概率公式，可以得到全期望公式：$\mathrm{E}[\mathrm{X}|\mathrm{A}] = \sum_{i=1}^{k} x_i \mathrm{P}(\mathrm{A}_i|\mathrm{A})$

Conjugate

- 有N堆石子，第i堆石子有A_i个，每次等概率随机选择一个石子把其所在的那一堆全部丢掉
- 期望多少次之后，第1堆石子被丢掉
- $N \le 10^6$
- $A_i \le 10^9$

Conjugate

- 对于每一堆，考虑其在第1堆之前被丢掉的概率
- 根据期望的线性性，求和就是答案
- 这个概率显然是$\frac{A_i}{A_i+A_1}$

湖人

- 有N个数1到N，每次会随机选择一个还未被删除的x，把包括其在内的所有倍数全部删去
- 求期望多少次后所有数都被删除
- 读入质数作为模数
- $1 \le \mathrm{N} \le 10^7$
- $1 \le \mathrm{N} \le 10^9$

湖人

- 考虑每个数被自己删掉的概率，根据期望的线性性求和就是答案
- 对于n，这个概率显然是$\frac{1}{\mathrm{d}_0(\mathrm{n})}$
- $\mathrm{N} \leq 10^9$加个min25筛就可以了

Game with Marbles

- 袋子里有R个红球，G个绿球和B个蓝球。每回合随机等概率从袋子中取出一个球
- 如果是红球，就扔掉
- 如果是绿球或蓝球，就放进袋子里
- 已知当前刚好拿出了第K个蓝球，求期望经过的回合数
- $1 \leq R, G, B \leq 10^9$

Game with Marbles

- 分成三部分，期望拿出了多少红球、蓝球、绿球，直接相加
- 已知拿出了K个蓝球，那么期望拿出K个蓝球
- 算期望摸到了多少绿球：考虑从什么都没有开始，第一次摸到蓝球之前期望摸到了多少绿球。这时候显然可以不考虑红球，这是个经典概率模型，可以考虑无穷级数求和，也可以直接列方程，$E = \frac{G}{B+G}(E+1)$，解得$E = \frac{G}{B}$
- 那么每摸一个蓝球之前期望摸了$\frac{G}{B}$个绿球，所以一共期望摸了$\frac{KG}{B}$个绿球

Game with Marbles

- 下面考虑算期望摸了多少个红球
- 这时候可以对每个红球单独计算摸出了它的概率
- 和前面一样，只考虑所有蓝球和这个红球。这时候相当于，如果这个红球没有被拿出来过，那么每次摸到所有蓝球和该红球的事件，都是选择去摸了蓝球，这个概率是$\frac{B}{B+1}$。如果不太能理解这个事情，就完整写出条件概率的式划一下就好了
- 那么每次摸蓝球都是进行了一个这样的事件，所以最后红球没被拿出来的概率是$(\frac{B}{B+1})^K$

游戏

- Source：UOJ#299 CTSC2017Day1T3
- 小R和小B在玩游戏，一共会玩N局
- 给定首局两个人分别获胜的概率。当第$\mathrm{i}-1$局小R获胜时，第i局小R获胜的概率为$\mathrm{P_i}$；当第$\mathrm{i}-1$局小B获胜时，第i局小B获胜的概率为$\mathrm{Q_i}$
- 会M次动态插入或删除一些已知的比赛结果，求小R获胜总局数的期望
- $1 \leq \mathrm{N}, \mathrm{M} \leq 2 \times 10^5$

游戏

- 根据期望线性性，同样是考虑每个未被确定的游戏的胜率
- 感性理解一下，显然每个游戏只和两边已确定的结果有关。所以相当于根据已确定的结果划分了若干个区间。每次插入删除就是删去旧区间加入新区间
- 假设X两边的已知事件是A和B，根据条件概率，$P(X|(A\cap B))=\frac{P(X\cap A\cap B)}{P(A\cap B)}=\frac{P((X\cap B)|A)P(A)}{P(B|A)P(A)}=\frac{P((X\cap B)|A)}{P(B|A)}$。用线段树维护矩阵就可以很方便地求出分母和分子的总和了

AGC035D

给定一个长度为 N 的数列 A，每次操作可以选择其中连续的三项，权值依次为 a, b, c，将其替换为 $a+b, c+b$ 这两项。

求经过操作最后剩下的两个数的和的最小值。

$$2 \le N \le 18, 0 \le A_i \le 10^9。$$

AGC035D

- 首先，操作可以看做是选择不是两边的一个数，将它删去并把它加到左右两项上
- 不妨考虑倒序考虑整个过程，同时最后的结果一定是每个数乘上一个系数的和。那么最开始，A_1和A_N的系数都是1
- 设$F[l][r][x][y]$表示，当前未被删除的数两端是l和r，中间的数已经被删除过了。A_l对答案贡献的系数为x、A_r对答案贡献的系数为y。转移考虑枚举第一个被加入（原过程中最后一个被删除的数为A_m），那么这时候代价就是$F[l][m][x][x+y]+F[m][r][x+y][y]$，注意这里还要扣掉被重复计算的$(x+y)\times A_m$

AGC035D

- 分析一下这个做法的时间复杂度。考虑从初始状态$x = y = 1$开始，会向下走不超过N层，每次只能转移到两个状态，所以(x, y)的个数是$O(2^N)$的
- 粗略分析一下，时间复杂度是$O(2^N \times N^3)$。精细分析可以得到更好的结果

CF765F

- 给定一个长度为N的序列，有M次询问
- 每次给定一个区间$[l, r]$，求$\min\limits_{l \le i < j \le r}\{|A[i] - A[j]|\}$
- $N \le 2 \times 10^5 \quad M \le 3 \times 10^5$

CF765F

- 考虑所有$i < j$且$A[i] \leq A[j]$，可能产生贡献的对(i, j)
- 枚举右端点r。考虑所有的x，满足$A[r] - A[x]$可能会产生贡献
- 把这样的x拿出来，$r > x_1 > x_2 > \cdots > x_M$
- 显然$A[x_1] < A[x_2] < \cdots < A[x_M] \leq A[r]$
- 而且$A[r] - A[x_{i+1}] < A[x_{i+1}] - A[x_i] \Rightarrow 2(A[r] - A[x_{i+1}]) < A[r] - A[x_i]$
- 所以每次$A[r] - A[x_i]$折半，直接用数据结构找到不超过$O(\log_2 W)$个x即可

operation

- 有一个n个点m条边的无向图，结点从1到n标号，点u的初始权值为w_u。对一个节点u进行操作定义为：将你的得分（初始为0）加上$\sum_{(u,v)\in E} w_v$，并将w_u除以2
- 给定一个长度为k的结点序列$s_1, s_2, \ldots, s_k$，在一轮操作中，你需要依次对 $s_1, s_2, \ldots, s_k$ 进行操作。你想知道，在进行了无限轮操作后，你得分的极限是多少
- 显然答案一定可以表示成P/Q的形式，你需要输出$P \times Q^{-1}$在模998244353意义下的值。特别的，如果得分并不收敛，输出-1
- $n, m, k \le 2 \times 10^5$

operation

- 只输出一轮操作后的得分怎么做?
- 度数分块，对于度数超过和不超过$\sqrt{m}$的点分别处理

operation

- 设在某一轮开始时，点u的权值是w_u。那么显然这一轮u对答案的贡献可以表示成$k_u \times w_u$的形式
- 假设u在序列中出现了Cnt_u次，则在下一轮对答案的贡献是$k_u \times w_u \times \frac{1}{2^{Cnt_u}}$
- 当$k_u \times w_u \neq 0$且$Cnt_u = 0$时，得分显然不收敛
- 否则，$1 + \frac{1}{2^{Cnt_u}} + \frac{1}{2^{2Cnt_u}} + \frac{1}{2^{3Cnt_u}} + \cdots = \frac{2^{Cnt_u}}{2^{Cnt_u - 1}}$
- 求每个点的k_u可以用前面提到的度数分块的做法

中美合拍

- 给定一个方程
- $x_1 \oplus x_2 \oplus \cdots \oplus x_n = k$
- 其中$\oplus$表示的是二进制异或操作，相信各位选手都很熟练了。
- 请求出该方程有多少组非负整数解，对于任意$1 \le i \le n$均满足$L \le x_i \le R$
- 有Q个这样的方程，希望你都能给出正确的答案
- 这个答案可能很大，请你求出对10^9+7取模的结果
- $0 \le Q \le 1000 \quad 1 \le n \le 10^9 \quad 0 \le L \le R < 2^{30} \quad 0 \le k < 2^{30}$

中美合拍

- 考虑$\mathrm{L} = 0$，即每个变量没有下界限制怎么做
- 一个方法是每次取最大值讨论，但这个方法不是很好，换一个思路
- 把所有上界的最高位提出来，分情况讨论，需要写一个dp

中美合拍

- 当存在下界限制时，很自然地想到容斥
- 考虑用和$L = 0$类似的方法做去统计贡献
- 具体而言，在dp的时候可以直接把带上了容斥系数的贡献算上去
- 这个dp显然可以用矩阵乘法优化，也可以用其他方法做到更优复杂度，但由于常数因子所以实现效率不明显且没太大意义
- 单次询问的时间复杂度是$O(D^3 \log_2 n \cdot 30)$

CF704C

给出一个形如这样的式子

$$(v_{1,1} \vee \ldots \vee v_{1,k_1}) \oplus (v_{2,1} \vee \ldots \vee v_{2,k_2}) \oplus \ldots \oplus (v_{n,1} \vee \ldots \vee v_{n,k_n}) = 1$$

和m个布尔变量x_i

其中每个$k \leq 2$，每个$v_{i,j}$都是一个是$x_{a_{i,j}}$或者$\neg x_{a_{i,j}}$

求解的数量（模$1e9+7$）

$1 \leq n, m \leq 100000$

$k_i \leq 2$

保证每一个变量最多出现两次

CF704C

- 考虑将一个将每一个包含两个变量子句（每一个括号）中的看成连一条边。那么，整张图就是若干个环和若干个链构成的
- 每个联通块是独立的。对于每个联通块分别算个答案，最后用一个大的dp合并起来就可以了
- 对于链就从端点开始往另一个端点做dp，设$F[i][a][b]$表示dp到第i个点，上一个变量取a、异或和为b的方案数。转移枚举当前变量是0还是1
- 对于环就先枚举某一个点的取值，然后破环成链做上面的dp
- 需要注意特殊情况，比如一个点连了两条自环。时间复杂度$O(N)$

对称性原理-翻折法

- 对称性原理在OI中最多的应用就是翻折法
- 来看这么一个例子：要找两条从(1,1)走到(N, M)的路径，满足这两条路径除了起点和终点不相交，计数方案数
- 先拆成两个起点到终点的路径
- 假设它们两独立，那么方案数就是直接相乘
- 对于相交的情况，在第一次相交的位置翻折，就相当于交换了起点（或者交换了终点），方案数还是直接相乘
- 两个结果相减就是答案

对称性原理-翻折法

- 再来看一个例子：计数从(0,0)走到(N, M)的方案数，但是需要时刻保证走到的点(x, y)满足$x \geq y$
- 相当于不能经过一条直线
- 对于经过一条直线的情况翻折，就变成了从起点到终点关于这条直线对称点
- 答案就是两个组合数相减

卡特兰数

- 卡特兰数有很多组合意义
- 括号序列计数
- 出栈序列计数
- 二叉树计数
- 凸多边形划分计数
- 括号序列和二叉树的一一对应：把二叉树从根一直往右走的路径拿出来，每个点打一个括号，括号里面的内容描述左儿子的子树
- 显然n个点的二叉树和$2n+1$个点的（每个点要么满儿子要么是叶子）二叉树一一对应

卡特兰数

- 考虑括号序列计数，实际上可以放在二维平面上，每次向右走就是左括号、向上走就是右括号，然后就是前面那个计数问题
- 求得卡特兰数的通项公式$\mathrm{Catalan}_{\mathrm{n}} = \frac{\binom{2\mathrm{n}}{\mathrm{n}}}{\mathrm{n}+1}$

概率论

- Source： BZOJ4001 TJOI2015
- 求N个点二叉树叶子数的期望
- $\mathrm{N} \leq 10^9$

概率论

- 对于一个叶子，可以用一个pair来描述：去掉该叶子后的二叉树，在该二叉树的哪个位置添加该叶子
- 每个pair也对应了唯一的叶子。所以考虑计数这个pair
- 去掉该叶子的二叉树有$N-1$个点，数目为$\mathrm{Catalan}_{N-1}$
- 对于一个$N-1$个点的二叉树，考虑有多少个空位可以放。总共有$2\times(N-1)$个空位，但是有$N-2$个点已经占据了一个空位，所以有N个空位可以添加叶子
- 两个方案数相乘，再除以$\mathrm{Catalan}_N$，化简一下发现答案是$\frac{N\times(N+1)}{2\times(2\times N-1)}$

Short LIS

- Source： Japan Alumni Group Summer Camp 2018 Day 2 K
- 计数有多少个长度为N的排列，满足：
- 最长下降子序列不超过2
- $\mathrm{A_x = y}$
- 这里有Q组询问
- $1 \le \mathrm{N, Q} \le 10^6$

Short LIS

- 对于一个排列“最长下降子序列不超过2” 这个条件，有很多等价的描述：
- （NOI 2018 Day1T2）冒泡排序次数达到下界
- 可以划分成不超过2个子序列，每个子序列单调上升
- 对于每个元素均满足：要么其前面所有数都比它小，要么比它小的数都在该元素的前面
- 还有等会做这道题的时候提到的一个转化

Short LIS

- 满足最长下降子序列不超过2的排列，一定与一种前缀max序列一一对应
- 因为考虑把前缀max序列中有用的值（就是单调栈中的值，也就是每一个连续段的开头）插入进序列时，为了使得最长下降子序列不超过2，就会产生从此处开始小于该数的所有数一定递增排列的限制
- 那么可以观察到，序列是这么一个形状：所有在前缀max上产生贡献的点形成了一个递增序列，剩下的数也形成了一个递增序列
- 显然一个最长下降子序列不超过2的排列一定对应了其本身的前缀max序列
- 而一个前缀max序列没有对前缀max产生贡献的位置填的数也是确定的，唯一对应了一个最长下降子序列不超过2的排列

Short LIS

- 分情况讨论：
- $y \geq x$，那么该点一定在前缀max上。考虑反证，若其没有对前缀max产生贡献，那么x前面一定有一个大于y的数，同时x后面没有小于y的数，否则存在长度为3的下降子序列，也就是所有小于y的数都在x之前，那么至少需要$1 + (y - 1) = y$个位置，而$y \geq x > x - 1$，x前面的位置不够，矛盾
- $y < x$，那么该点一定不对前缀max产生贡献。这是显然的
- 对于$y < x$的情况，考虑A的逆置换$A^{-1}[A[i]] = i$，可以转换成另一种情况，也就是说只要考虑$y \geq x$的情况

Short LIS

- 考虑一个合法的前缀max序列满足的条件：
- （1）$1 \le \max[i] \le n \quad \max[n] = n$
- （2）$\max[i] \le \max[i+1]$
- （3）$i \le \max[i]$
- 同样放到平面上考虑。固定$A[x] = y$等价于确定了一部分路径，两边分别计数即可。计数也是用翻折法
- 那么预处理复杂度是$O(N)$，单次询问复杂度是$O(1)$

多次翻折法

- 这是我自己瞎取的名字
- 一般的翻折法可以解决不经过一条直线的问题。多次翻折法用于解决不经过两条平行直线的问题，称之为上界和下界
- 第一次先往上翻，然后翻下来，以此类推，方案数加到答案里面的系数是$+1, -1, +1, \ldots$
- 第二次先往下翻，然后翻上去，以此类推，方案数加到答案里面的系数是$+1, -1, +1, \ldots$
- 可以发现，一个不合法的方案被计算的次数是$1+0$或是$0+1$，是前者还是后者取决于它是先抵到上边界还是下边界，但反正都是1

count

- Source：UOJ#424 IOI2019集训队作业by周驿东
- 根据笛卡尔树的性质直接转化，得到正常题意
- 计数有多少大小为N的二叉树，满足从根节点走到任意一个节点向左走的次数不超过M
- $1 \leq \mathrm{N} \leq 10^6$

count

- 首先有一个$O(NM)$的暴力dp。考虑按照先序遍历的顺序遍历这颗二叉树；那么在每个点处要么向左走，要么跳到其某个祖先（包括自己）处然后往右走
- 那么dp的时候只需要记录当前是先序遍历的第几个点，以及向左走了多少次。转移用前缀和可以优化到线性

count

- 优化dp可以考虑组合意义，即计数有多少个长度为N的非负整数序列A，满足：
- $0 \le A_i \le M$
- $A_1 = 0$
- $\forall 2 \le i \le N, A_i \le A_{i-1} + 1$
- 特判掉$M = 0$的特殊情况，对于第三条限制，常见的变换手段是令$P_i = i - A_i$

count

- $i - M \le P_i \le i$
- $P_1 = 1 \quad P_{N+1} = N + 1$
- $P_i \ge P_{i-1}$
- 即从$(1,1)$走到$(N + 1, N + 1)$，每次只能往右走和往上走，把每个x处的最高点当作P_x的值，那么不能经过$y = x - (M + 2)$和$y = x + 1$两条直线的方案数
- 多次翻折法

Jiry Matchings

给一棵边权树，$f(i)$ 表示 i 条边的最大权匹配，求出 $f(1), f(2), \ldots, f(n-1)$.

$n \leq 2 \cdot 10^5$，时限 6s.

Jiry Matchings

- 考虑如何做两个凸函数的$\max +$卷积，可以考虑差分后归并
- 再考虑一个经典的树上分治FFT做法，即对轻儿子做分治FFT，对所有重链做分治FFT。把上面那个做法套上去就可以了

CF603E

N 个点，M 次操作，每次操作加入一条带权无向边，然后询问当前图是否存在一个生成子图，满足所有点的度数都是奇数；如果有，输出所有满足条件的生成子图中最大边权的最小值 (没有则输出-1)。

对于所有数据，保证 $2 \leq N \leq 10^5, 1 \leq M \leq 3 \times 10^5$, 边权 w_i 满足 $1 \leq w_i \leq 10^9$。

时间限制：$4s$

空间限制：$256MB$

CF603E

- 存在方案等价于每个联通块点数都为偶数。证明考虑对每个联通块任意取一颗生成树，非树边任意选，用树边来调整度数
- 显然加的边越多越有可能满足方案。这也可以从条件来解释，因为同奇偶性的联通块合并得到偶联通块，不同奇偶性的联通块合并得到奇联通块
- 用LCT维护最小生成树，同时尽量删除边权较大的边