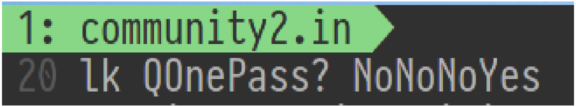
杂题选讲

lk

2024 年 7 月 7 日

按照惯例来个自我介绍

- CTSC2017 Fe
- WC2019 Cu
- HNOI2019 挂的分比拿的分多 45 & 计数题爆栈哥
- ISIJ2019 Ag (40+ac+ac)
- NOI2020 Ag (day1 50+72+56)
- WC2021 64+100+80
- THUSC2021 50+100+100+100
- NOI2021 day2 24+100+36
- CTT2021 前 3 天 T1 得分分别为 50.36,45,30
- ICPC 获得了亚军、亚军、亚军、亚军和银牌，并在 World Finals 中写三题挂 13 发。



Fabulous Fungus Frenzy

- 给出一个 $n \times m$ 的矩阵，每次可以交换一对相邻元素，顺时针旋转一个 2×2 的子矩阵，或者选择一个预设矩阵覆盖到某个子矩阵上。(给定 k 个预设矩阵)
- 给一个方案，将矩阵转成另一个矩阵，要求操作次数不超过 $4 \cdot 10^5$ 且预设操作次数不超过 400。
- $n, m \leq 20$

题解

- 注意到覆盖是不可逆的，那么倒着考虑问题：每次覆盖操作撤回相当于把矩阵全变成通配符。
- 这样我们其实不用关心字符的具体位置，只用考虑每种字符的个数，每次撤回一个盖章操作相当于把一些字符变成通配符，那么每次只有当能增加通配符个数时我们再加上这个操作，这样最多会进行 $nm \le 400$ 次。对于每次盖章操作，我们只需要把外面的字符手动推进来，推进来需要 $n+m$ 次，总共是 $\le nm(n+m)$ 次，从而最终总操作数是 $n^2m^2(n+m) \le 200000$ 次。

Pumping Lemma

- 给定两个串 s, t，求有多少个三元字符串组 (x, y, z) 使得存在 k 满足 $s = x + y + z, t = x + y^k + z$。
- $|s| < |t| \leq 10^7$.

题解

- 先枚举 $x+y$，那么串就会被拆成 $x+y, y^k, z$ 三部分，我们考虑数有多少种 y，也就是求 y^k 有多少种循环节满足也是 $x+y$ 的后缀。
- 直接求最小循环节很困难，考虑结合上 “是 $x+y$ 的后缀”，就是要求循环节 $|y| \le LCS(s_{1\ldots k}, s_{k+1\ldots k+(n-m)})$。注意到中间这部分串长是固定的了，然后发现从 k 转移到 $k+1$ 的时候如果 $s_{k+1} = s_{k+(n-m)+1}$ 则 LCS 会加一 (可行的 $|y|$ 可能多一个)，否则清零。然后我们可以用 hash 判断这个新加的 y 是否可行。

Vabank

- 你需要猜一个数 M，你初始有个分值 $P=1$，每次可以猜 x，会告诉你 $x \le M$ 还是 $x > M$，如果 $x \le M$ P 会 $+x$，否则 $-x$，如果 $P < 0$ 就寄了。
- 操作限制 105，$M \le 10^{14}$

题解

- 先从小到大枚举 M 的位数，这个位数不超过 47。
- 第一次寄的时候，操作了不超过 46 次，且此时 $P=0$。
- 假设现在可知答案在 $[2^k,2^{k+1})$，先询问两次 2^k 补充 P。
- 然后我们可以考虑一个类似三分的思路：分成三段，先询问 $mid1$ 再询问 $mid2$，如果 $mid1$ 就寄了则多询问一次 l 补充 P，这样每次之后 P 最多会减少 $\frac{1}{3}$ 段长，且此时 r 也会减少至少 $\frac{1}{3}$ 段长，所以可以发现 P 一定够用。
- 对于 $[2^{46},10^{14}]$ 的情况，一开始不用补 P，然后一样做就可以了。
- 但是在 $[2^{45},2^{46})$ 中操作次数是 $29*2+2+46=106(\lceil\log_3 2^{45}\rceil=29$，多了一次。
- 观察到当 $l+1=r$ 时只用询问一次，就能减少一次 $(2\cdot 3^{28}\geq 2^{45})$，就卡进上界了。

机器蚤分组

- 给定一个字符串 S，q 次询问对于一个子串 $S_{l\dots r}$ 的**所有**本质不同非空子串，最少能划分成多少组，使得每一组都形如 $a_1, a_2, \dots$ 且 a_i 是 a_{i+1} 的子串。
- $|S|, q \le 10^5$

题解

- 首先最小链覆盖 $=$ 最长反链。
- 然后结论 1：最长反链答案 $=$ 同一长度的不同子串数最大值。
- 原因：如果存在一个反链串长不全相同，设最短的串长为 len，如果有一个串 $s_{l..l+len-1}$ 在反链中，且 $s_{l-1...l+len-2}$ 或 $s_{l+1...l+len}$ 不在，就可以变成 $s_{l-1...l+len-1}$ 或者 $s_{l...l+len}$，反复进行操作最后会让所有串长相同。
- 结论 2：答案 $=k$ 当且仅当所有长度为 $n-k+1$ 的子串不相等。
- 原因：如果有长度为 $n-k+1$ 的子串相等，那么对于所有串长 len 满足 $len \leq n-k+1$，这个串长的不同子串数不会超过 $(n-len+1)-(n-k+1-len+1)=k-1$。
- 所以答案就是 $|S|-\max_{i<j} lcp(S[i:],S[j:])$.
- 考虑怎么快速解决多组询问，考虑在后缀树上启发式合并，每次合并两个后缀集合 S_1,S_2 的时候只会有 $x\in S_1, y\in S_2$ 且 x,y 在 $S_1\cup S_2$ 中相邻的这种 pair (x,y) 有影响。也就是只有不超过 $\min(|S_1|,|S_2|)$ 对，所以总共的 pair 数不超过 $O(n\log n)$，然后对询问离线，对右端点扫描线，用线段树维护左端点的答案即可。

Favorite Game

- 有 n 个传送门，0 时刻你可以选择一个整点到达，每个时刻可以四个方向选一个移动一步。
- 如果走到传送门，则会激活，激活后任意时刻可以瞬间传送到任意激活的传送门。
- 有 m 个任务，第 i 个要求 t_i 时刻恰好在 (x_i, y_i)，求最多能完成几个任务。
- $n \leq 14, m \leq 100$

题解

- 容易想到的 dp: $dp_{s,i,j}$ 表示激活了 s，当前在 i，完成了 j 个任务的最小时间，状态数就 $2^n(n+m)m$ 了。
- 考虑把 i 是门和 i 是任务点分开：对于门的情况，因为可以随意互相传送，我们只用考虑它在门，即 $f_{s,j}$ 表示当前在一个门的位置完成了 j 个任务的最小时间。
- 而对应的，任务点的情况我们不用考虑最小时间，因为时间一定是 t_i，从而只用考虑 $g_{s,i}$ 表示当前时间 t_i，开了 s 集合门，最多做了几个任务。
- 这样状态数就健康了，总复杂度是 $O(2^nm^2)$.

Four Vertices

- 一个边带权无向图，支持加边删边，每次操作完后，需要找 4 个不同的点 a, b, c, d 使得 a, b 连通，c, d 连通，且 $dis(a, b) + dis(c, d)$ 最小。
- $n, m, q \leq 10^5$

题解

- 易知最优答案一定是一个点出发的三条边或两个不交的边。
- 前一个 trivial，只考虑后面一个。
- 然后可以发现，要么是最短边加某条边，要么是两个点分别出发一条边。
- 有个结论是，如果一条边不是它某个端点的前三小的边，则它一定没用。
- 于是维护下每个端点的前三小和这些合并到一起的 set，然后两个 case 都很好枚举了。

Say Hello to the Future

- 有个长度为 n 的序列 a，需要把这个序列划成一些区间，使得 i 所在区间的长度是至少是 a_i(也即 len≥max)，记 $f(a)$ 为划分方案数。
- 对于所有 $1 \le i \le n$，把 a 第 i 个位置变成 1 其他位置不变后，$f(a)$ 的新值。
- $n \le 2 \cdot 10^5$，$\mathrm{mod} 998244353$。

题解

- 先考虑原序列的答案怎么求，首先可以 dp f_i 为 $1 \dots i$ 的划分数，转移考虑分治跨过 mid 的转移：
  $r - l + 1 \geq \max(mxl, mxr) \iff r \geq l + mxl - 1 \land l \leq r - mxr + 1$.
  二维偏序，树状数组就是 $O(nlog^2 n)$
- 考虑改个位置之后的答案，先类似的算出每个后缀的划分数，然后类似的在分治树上考虑，修改 x 可能会修改一些 mxl 或者一些 mxr，但易知每个位置只会被一个 x 修改，这样我们就还能类似的二维偏序做。

霸占排行榜

- 给定一棵 l 个点的 trie 树，和 n 个串 $s_1 \dots s_n$。
- 给定 $a_1 \dots a_m$，令 $T = s_{a_1} + s_{a_2} + \cdots + s_{a_m}$，对于 trie 树上每个节点，求对应的串在 T 中出现的次数。
- $n \le 100, \sum s_i \le 3 \cdot 10^6, m \le 10^6, |l| \le 10^5$，字符集大小 $4 \cdot 10^6$.

题解

- 考虑先对每个串 s_i 和节点 u_j 求出 $len_{i,j}$ 表示 u_j 最多加上 s_i 的多长前缀还不会失配，$end_{i,j}$ 表示此时会走到哪
- 然后考虑求出 $dir_{i,j}$ 表示 u_j 匹配完 s_i 会结束在哪，考虑第一次失配的时候，假设 $k = fail_{end_{i,j}}$，如果 $dep_k \le len_{i,j}$ 则可知前面的部分全没用了 (即 $dir_{i,j} = dir_{i,1}$)，否则剩下的部分等价于从 $fa_k^{len_{i,j}}$ 开始跑，也就是 k 在 **trie** 树上的 $len_{i,j}$ 级祖先。
- 求答案的时候也类似，先走到 $end_{i,j}$ 跳到 $k = fail_{end_{i,j}}$，如果 $dep_k \le len_{i,j}$ 则剩下的部分就是对 $match(rt, s[k' :])$ 做后缀加。否则剩下的部分相当于从 $fa_k^{len_{i,j}}$ 开始。从下往上做个树上差分就行。
- 那么问题在于怎么求 $end_{i,j}$ 了，考虑先在 trie 树的重链上 exkmp 求出最长前缀，剩下的部分就是求某个轻子树 a 里后缀 b 的匹配情况。我们可以离线下来，对每个子树按照后缀的字典序升序一个个算，每个匹配点从上一个匹配点开始修改，每个点最多会被访问一次。
- 总复杂度 $O(nl + S + m + l \log l + l \log w)$.

Yet Another Maximize Permutation Subarrays

- 给一个排列，可以交换某两个位置，最大化交换后的子排列数组个数。子排列数组的定义为，是一个子串，且其中的元素是 1.. 长度的一个排列。
- $n \leq 10^6$
- a.k.a. CTT Day3T1
- 「有兴趣的同学可以想想第 k 大，万一什么时候我又出来祸害你们了 x」

题解

- 考虑这个排列 p 的逆排列 q，也即位置数组，子排列数组相当于位置数组上一个前缀使得是连续段。
- 注意到当我们交换 $x, y(x < y)$ 时只会对 $[x, y)$ 区间产生影响，然后先记 $q_1, \dots, q_i$ 的值域为 l_i, r_i (即 min 和 max)，如果 $q_x \neq l_i \wedge q_x \neq r_i$ 则 $1 \dots i$ 一定不是连续段，则我们只用考虑 $q_x = l_i$ 或 $q_x = r_i$ 这样的位置，且换了还是连续段 (或者本来不是变成是) 的 y 也只有最多两个，于是换了之后能新增的连续段只有 $O(n)$ 对，而换了之后减少的部分比较好算，这样对每个可能增加的 x, y 判一下就成。
- 这里需要特判一下 $x = 1$，只需要对于 $2 \dots i$ 算出加谁可以变成连续段就行。

Easy Diameter Problem

- 给一棵树，每次可以选一条直径，将其某一个端点删了，最终会将树删光，求删除顺序个数 $b \mod (10^9+7)$
- $n \leq 300$

题解

- 对于所有直径一定有一个公共中心点 (可能是边中间的点)，记 d 为直径长度 r 为半径长度 $\frac{d}{2}$，x 为中心，那么删的直径端点就是所有和 x 距离 $= r$ 的点，直到只剩一棵子树还有这样的点为止，此时中心会朝子树方向移动 $\frac{1}{2}$ 条边,r 也会 $-\frac{1}{2}$. 我们对于这个子树里的直径端点，可以先不管删了谁，只考虑删了几个留了几个把它们用组合数合并到前面，最后真删的时候再管删了谁。
- 于是我们 dp 一下，$f_{x,r,fa,c}$ 表示现在中心是 x，半径是 r，x 的父亲是 fa(或者说从 fa 转移过来的)，以 fa 为根 x 的子树里距离 $= r$ 的点还剩余 c 个的方案数。容易注意到，虽然看上去是四维其实状态数只有 $O(n^2)$.
- 转移就是转移到某个距离 $\frac{1}{2}$ 的点，$r - \frac{1}{2}$, 然后我们可以 $O(n)$ 枚举新的 c 转移，复杂度就是 $O(n^3)$.

Pink Floyd

- 有个竞赛图，边有粉色和绿色，你知道所有粉色边的方向。
- 现在你可以询问最多 2n 次某个绿色边的方向，需要找到一个点满足可以从这个点通过全程走的边颜色一样的路径到达每个点。
- adaptive.
- $n, m \leq 10^5$

题解

- 没粉边的情况维护外向树的根，每次找两个合并就行。
- 有粉边的情况先缩点，拓扑排序，每次把所有入度为 0 的点拿出来任选两个合并，然后获得入边的那个点就删掉，把新的入度为 0 的点加进来。
- 如果只剩一个点的时候，还没加进来的点一定能被粉色边到达，其它的点都在这个点的外向子树里，所以找到了。

Communism

- 有一个串，每次选一种在串里出现了的字符 x，设出现位置为 $a_1, \cdots, a_m (a_i < a_{i+1})$，若 $k(a_m - a_1 + 1) \le m$，则可以将所有 x 变为另一个在串里出现了的字符 y。
- 求有哪些字符可以使得最后串里所有字符都是它。
- $n \le 5000, k = \frac{a}{b}, 1 \le a \le b \le 10^5$，字符集大小 20。

题解

- 显然合并结构是一棵树。首先我们知道哪些字符子集合并在一起之后可以转化到别的字符。
- 记 f_S 为 S 集合是否能被合并到一起，同时能否转化给别的字符。
- 设 g_S 为 S 集合是否能被拆分成一些能转化的子集。
- 直接子集 dp 是 3^{Σ} 的，肯定过不了。
- 考虑如果拆分成的某两个集合相交了，一定可以把其中一个先合并到另一个而不影响能否转化。从而我们转移的时候只用按照 ri 排序之后取一个后缀或前缀即可。

Parity Game

- 给定一个 01 序列 a 和一个 $t \in \{0,1\}$, Alice 和 Bob 可以轮流进行操作：每次选相邻两个位置，替换成和或积，直到只剩一个数。
- Alice 先手，如果最后的数 $\equiv t \pmod 2$ Alice 赢，否则 Bob 赢。
- 交互题，你需要选择你要当 Alice 还是 Bob，然后赢下来 (相当于给出你的必胜策略)，$n \leq 500$。

题解

- 如果走最后一步的人要求是变成 0，则他必胜，因为最后一步剩下任意两个数都一定能变 0。
- 如果走最后一步的人要求是变成 1，那么必胜的条件为：把所有连续段 1 缩成单个 $0/1$(为连续段长度的奇偶性) 之后，1 的个数至少要不小于 0 的个数。
- 为什么呢？因为走最后一步的人输了当且仅当剩下两个数全是 0。所以他会希望最大化 1 的个数 -0 的个数，而另一个人会最小化这个个数，考虑一下任意操作的贡献就可以了。
- 由于 n 不大，具体操作的时候可以直接枚举然后判操作完之后是否还是必胜态，$O(n^3)$ 就能过。

企鹅游戏

- 给定 n 个两两不同的串 $s_1 \dots s_n$，q 次询问一个串 t，令 c_i 为 s_i 在 t 中出现的次数，求 $\sum 3^{i \cdot c_i} \bmod 2^{32}$.
- $|s_i|, |t|, n, q \le 2 \cdot 10^5, \sum |s_i|, \sum |t| \le 2 \cdot 10^6$.

题解

- 记 $L = \max(\sum |s_i|, \sum |t|)$.
- 先建 AC 自动机，对于每个节点求出 fail 树上最近的能代表一个子串的节点，然后询问的时候每次对着这条链往上跳复杂度就是 $O(L\sqrt{L})$(由于串的不同长度只有 $\sqrt{L}$ 种)。
- 然后注意到，一个 t 中出现的不同 s_i 个数的和是 $O(L^{4/3})$ 级别的 (对于长度 $\leq L^{\frac{1}{3}}$) 的串，每个长度的串总出现次数 $\leq L$，也即 $O(L^{4/3})$，对于长度 $> L^{\frac{1}{3}}$ 的串，每个最多会在 $L^{\frac{2}{3}}$ 个 t 中出现，且一共只有 $L^{\frac{2}{3}}$ 个这样的串)。
- 于是我们可以造一个复杂度是出现了的 s_i 个数的答案：在往上跳的时候打标记，如果遇上打了标记的点就停，然后差分求和的时候写个拓扑排序。

Extending Distance

- 有 nm 个点排成了一个 $n \times m$ 的矩阵，相邻两个点之间有条边，每条边有个权值。定义这张图的价值为第一列某个点到最后一列某个点的最短路 (min 最短路 for all 点对)。现在可以进行任意次操作，每次是选择一条边给边权 $+1$，求最少的操作次数使得价值增加 k。
- $n \times m \leq 500, k \leq 100$

题解

- 对偶，最后一行下面的部分设为源点，第一行上面的部分设为汇点，最短路就是对偶图上的最小割。
- 然后将原图中的边费用 $=0$，流量设为边权，同时加个费用 $=1$ 流量 $+\infty$ 的重边，就能把问题转成求流量 $D+K$ 的最小费用流了，显然费用为 0 的流量就有 D 了，可以先跑个 dinic，复杂度就是 $O(n^2m^2K)$。

To Make 1

- 给定一个常数 k，记 $f(x)$ 为将 x 不断除 k 直到无法整除位置之后得到的数。
- 有 n 个数，每次选两个数 x, y 删掉并将 $f(x+y)$ 加进来，直到最后剩下一个。
- 问是否能让最后的数字为 1(如果能，构造方案)
- $2 \le n \le 16, 2 \le k, \sum a_i \le 2000$，且保证初始所有数字不是 k 的倍数。

题解

- 对于一组解，每个数对最后的数的贡献都形如 $\frac{a_i}{k^{b_i}}$，其中 $b_i \geq 0$。
- 事实上只要确定一组满足条件的 b_i 就能还原一个方案，证明 (构造) 很容易。
- 从而我们只需要先确定一组 b_i，这时我们不需要考虑加会立刻除 k 了，就可以直接状压 dp 了。
- bitset 优化之后是 $O(2^n \sum a_i + \frac{n 2^n \sum a_i}{w})$ 的。

Dreamoon Loves AA

- 有一个长度为 $n+1$ 的 AB 串，首尾为 A，除此之外还有 m 个 A，现在希望修改恰好 k 个 B 为 A，使得所有相邻 A 之间距离的极差最小。
- $n \leq 10^{15}, m \leq 400000, k < n-m$

题解

- 题意等价于，有 $m+1$ 个间隔 b_i，你可以拆分间隔恰好 k 次，最小化极差。
- 首先有一个显然的事情：拆分一定是尽量均匀，即拆成 $\lfloor \frac{b_i}{x} \rfloor, \lceil \frac{b_i}{x} \rceil$。
- 然后假设 $\min, \max$ 是 L, R，有两个条件（设 $K = m+k+1$ 为总段数）：
- $\sum \lceil \frac{b_i}{R} \rceil \le K \le \sum \lfloor \frac{b_i}{L} \rfloor, \exists x, \lceil \frac{b_i}{R} \rceil \le x \le \lfloor \frac{b_i}{L} \rfloor$
- 容易发现这两个条件是必要条件，加起来也是充分条件。
- 第一个条件可以二分出 L 的上界 L_0 和 R 的下界 R_0。第二个条件对于 L_0, R_0 不满足的情况只需要修改 L, R 之一。
- 问题转化为集合里 L_0, R_0，有一些 pair$(L, R)(L \le L_0 \le R_0 \le R)$ 从每个 pair 里选一个扔进集合最小化极差。
- 枚举最小值即可。

Weighted Increasing Subsequences

- 有个序列 a，对于一个上升子序列 $a_{i_1}, a_{i_2} \ldots, a_{i_k}$，定义它的权值为 j 的个数，使得存在 $x > i_k$ 有 $a_{i_j} < a_x$.
- 求 a 中所有上升子序列的权值和 $\mathrm{mod}(10^9 + 7)$.

题解

- 考虑拆贡献，对每个 a_{i_j} 数子序列个数。
- 也就是先枚举 a_x，然后设 $> a_x$ 的最大位置为 a_y，只要 y 不在子序列里就行。
- 注意到 a_y 是严格后缀最大值，设下一个严格后缀最大值是 a_z，则 $a_z \leq a_x < a_y$，我们只要对这个子集 dp 一下即可，上面的部分可以扫描线下来，下面的也类似。
- 时间复杂度 $O(n \log n)$.

The End

Thanks!