2024 威海市赛题解

2024 年 9 月 28 日

Contents

Contents

赛前预测

- easy: EHL
- easy-mid: BG
- mid: DFJ
- mid-hard: AKMI
- hard: C
- 捧杯: 9+
- 金: 7+
- 银: 5
- 铜: 4

Contents

A: 大家的公因数 2

由题意可知，两个点有一个大于 1 的公因数时会连边，则我们可以枚举每个数对的因数，根据因数的情况进行连边。
这样建图的复杂度为 $\mathcal{O}(n^2\sqrt{a_i})$ 的，不能接受。
考虑不枚举数对，可以通过并查集维护每个数对应的连通块，在同一个连通块 i 中的数至少都有公因数 i。
暴力枚举每个数的因数时复杂度为 $\mathcal{O}(n\sqrt{a_i})$，本数据范围无法通过，对每个数进行质因数分解即可。
不需要显式建图，只需要记录下每个连通块中有哪些点即可。

A: 大家的公因数 2

对于查询 (u_i, v_i, w_i)，进行分类讨论：

- 若 u_i 和 v_i 不在同一个连通块，必然无法找到合法的路径；
- 若 u_i 和 v_i 在同一个连通块，由样例解释可知，我们可以经过同一个点两次，从而消除该点对异或和的影响。因此，问题转化为，给定若干数，能否从其中选出一些数使得其异或和为 w_i。线性基即可解决。

时间复杂度为 $\mathcal{O}\left(n\sqrt{\frac{a_i}{\ln a_i}} + q\log n\right)$。

冷知识：本题命制于 CCPC 网络赛前一天，第二天就来了一道线性基（

B: 抽卡

设 $dp[i][j]$ 为还有 i 个 A，j 个 B 后的期望猜对次数。

$$dp[i][j] = \begin{cases} dp[i][j-1] & i = j \\ \dfrac{i}{i+j} \times (dp[i-1][j] + 1) + \dfrac{j}{i+j} \times dp[i][j-1] & i > j \\ \dfrac{j}{i+j} \times (dp[i][j-1] + 1) + \dfrac{i}{i+j} \times dp[i-1][j] & i < j \end{cases}$$

复杂度 $\mathcal{O}(n^2)$。

B: 抽卡

我们假设 $a < b$
根据数学归纳法

$$ans = \frac{b}{a+b}(b-1+1) + \frac{a}{a+b}(b) = b$$

所以答案就是 $max(a, b)$。

C: 状态图

根据题意，我们可以轻松构建如题所述的状态转换图。
一个暴力的想法是，从所有节点开始，分别以高/低脉冲开始，在状态转换图上全跑一遍，可以在 $\mathcal{O}(n)$ 的时间内判断是否有解。
有解时，必然会形成若干的独立的环，取这些环长的 lcm 即是答案。
然而由于 n 过大，不能直接建图，甚至不能确定答案的范围。
因此，需要考虑答案所具有的性质。
有解的充要条件是：$n \perp a, n \perp b$，即 n 与 a, b 皆互质。
首先，由于一个周期需要保证从高（低）脉冲开始，回到高（低）脉冲，即 T 必然是偶数。
既然如此，直接一次性考虑两个脉冲的转换（高-低/低-高），则根据题意，转移为：
高-低：$x \to b(ax + c) + d \mod n$
低-高：$x \to a(bx + d) + c \mod n$

C: 状态图

事实上，在有解情况下只需要考虑其中一种转移即可，不妨取高-低型转移。

设 $f(x) = b(ax + c) + d = abx + bc + d \equiv kx + m \pmod{n}$

则我们所关注的是满足 $f^t(x) \equiv x (\bmod n)$ 的 t 的最小值。

展开上式，即

$$(k^t - 1)x + (k^{t-1} + k^{t-2} + \cdots + 1)m \equiv 0 \pmod{n}$$

若 $k \equiv 1 \pmod{n}$，即 $tm \equiv 0 \pmod{n}$，$t_{\min} = \frac{n}{(n,m)}$.

若 $k \not\equiv 1 \pmod{n}$，要使得同余式对于所有 $x = 0, 1, \cdots, n-1$ 皆成立，显然必要有

$$k^t - 1 \equiv 0 \pmod{n}$$

以及

$$(k^{t-1} + k^{t-2} + \cdots + 1)m \equiv 0 \pmod{n}$$

C: 状态图

由前一个式子得，t 必定是 k 在模 n 意义下的阶 $ind = ind_n(t)$ 的倍数，即

$$t = \omega \cdot ind$$

将该式代入后一个式子，并设 $s = k^{ind-1} + k^{ind-2} + \cdot + 1$，得

$$\begin{aligned}
&(k^{t-1} + k^{t-2} + \cdots + 1)m \\
\equiv\ & (k^{\omega\cdot ind-1} + k^{\omega\cdot ind-2} + \cdots + 1)m \pmod n \\
\equiv\ & [k^{ind}(k^{(\omega-1)ind-1} + k^{(\omega-1)ind-2} + \cdots + 1) + s]m \pmod n \\
\equiv\ & [(k^{(\omega-1)ind-1} + k^{(\omega-1)ind-2} + \cdots + 1) + s]m \pmod n \\
\equiv\ & \cdots \\
\equiv\ & \omega \cdot s \cdot m \\
\equiv\ & 0 \pmod n
\end{aligned}$$

C：状态图

解得 $\omega_{\min} = \frac{n}{(n, s \cdot m)}$，$t_{\min} = \omega_{\min} \cdot ind$，$t_{\min}$ 即为所求答案。
总的复杂度为 $\mathcal{O}(T\sqrt{n}\log n)$。
其中，求 $ind_n(t)$ 是 $\mathcal{O}(\sqrt{n} \cdot \log n)$ 的，因为要枚举 $\varphi(n)$（当然求 φ 也需要 $\mathcal{O}(\sqrt{n})$）的因子试阶；
而在模意义下求 $s = k^{ind-1} + k^{ind-2} + \cdot + 1$ 不能直接用等比数列求和公式，暴力求又是 $\mathcal{O}(ind) = \mathcal{O}(n)$ 的，无法接受，因此可以用矩阵快速幂的方法在 $\mathcal{O}(2^3 \log n)$ 的时间复杂度内解决。

D: 找数

首先我们要满足选出来的序列奇数位为奇数，偶数位为偶数，这样有两个限制搞得我们措手不及，所以我们考虑一下，如何通过构造使其简化。
因为我们每个数的奇偶性和位置的奇偶性相关，所以我们考虑选择排序后的序列 $a_{1...m}$ 我们对于第 i 位的数 $+i$ ，这样之后我们得到一个互不相同的长度为 m 的全偶数的递增序列。
然后我们问题就转化为构造这个新的全偶序列的方案数，即从 $1....n+m$ 中选出 m 个偶数的方案数

$$ans = C^{m}_{\lfloor\frac{n+m}{2}\rfloor}$$

E: 张灯结彩

本题同时为高职组的 D 题。
找规律题。
答案为

$$3 \times \sum_{i=1}^{n-1} (2i - 1) + 2(n - 1)$$

$\mathcal{O}(n)$ 可过，也可以 $\mathcal{O}(1)$。

F: 异或盒子 1

我们考虑对每个数的二进制，从小位到大位建 trie，对于 $+1$ 操作，其实就是前面的有连续的 $1 \to 0$，第一个 $0 \to 1$，在 01-trie 上就类似于 reverse 一下，所以我们从根节点开始每次直接交换左右子树，然后进入原本右儿子重复该操作。

维护一下每个点的 siz，在 reverse 的时候判断一下 $siz[ls[]]$ 与 $siz[rs[]]$ 的奇偶性来更新我们的总异或和。

每次我们只会翻转 $\mathcal{O}(\log)$ 个子树，复杂度 $\mathcal{O}(n\log n)$ 。

对于维护每一个位置都是什么数，数组维护一下就好了 (整体 $+1$, 单点修改，单点查询)。

总复杂度 $\mathcal{O}(n\log n)$

G: 最大拟合

简化版的题意就是选择一个最优的参数组

$$\theta(L,L;L),\theta(L,L;H),\theta(L,L;\$)$$
$$\theta(L,H;L),\theta(L,H;H),\theta(L,H;\$)$$
$$\theta(H,L;L),\theta(H,L;H),\theta(H,L;\$)$$
$$\theta(H,H;L),\theta(H,H;H),\theta(H,H;\$)$$

其中参数组之间有约束如

$$\theta(L,L;L)+\theta(L,L;H)+\theta(L,L;\$)=1$$

且上面的每项均非负。
优化目标等价为

$$\max_{\theta}\sum_{i=3}^{m+1}\log\theta(s_{i-2},s_{i-1};s_i)$$

G: 最大拟合

我们只需要考虑序列中 S_{i-2}, S_{i-1}, S_i 三元组出现的次数即可。
转化后，会发现等价于优化多组不等式，形如

$$\max_{p+q+r=1, p,q,r\geq 0} a\log p + b\log q + c\log r$$

a, b, c 就是某些三元组出现的次数。考虑 a, b, c 中存在一些 0 的情况，如 $a = 0$，此时令 $p = 0$ 显然最优，问题可以简化成更简单的形式；否则，考虑 $0 < a, b, c < 1$ 的情况，不等式的最大值在

$$p = \frac{a}{a+b+c}, q = \frac{b}{a+b+c}, r = \frac{c}{a+b+c}$$

处取得。
对每个不等式取最优，合计得到结果。复杂度 $\mathcal{O}(|S|)$。

H: 不是一道简单的构造题

对发布的错误 clar 表示抱歉（滑跪

对 k 分类讨论：

- 若 k 为偶数，则关于点 u 对称构造即可；
- 若 k 为奇数，在 u 点也构造一个即可。

综上，答案总是 `Yes`。时间复杂度 $\mathcal{O}(1)$。

I: 衡量距离

考虑所有边长 w 长度均为 1 的情况，题目退化为经典模型，最朴素的做法就是将有向图的 01 矩阵，进行 k 次幂可得到路径长度为 k 的可达矩阵。之后用 bitset 去优化 $0,1$ 矩阵乘法。考虑原始的矩阵乘法

$$for(k,i,j)\ \ c[i][j]| = a[i][k]\&b[k][j]$$

前第三层循环中 $a[i][k]$ 的值不变，且只有为 1 的时候有用，因此可用 $bitset$ 优化转变

$$for(k,i)\ \ if(a[i][k])\ \ c[i] = c[i]|b[k]$$

时间复杂度为 $\mathcal{O}(\frac{n^3}{64} * \log k)$

I: 衡量距离

再考虑边长不为 1 的情况，最直观的方法就是将长度为 w 的边，拆为 w 个长度为 1 的边。
即在之间加入 $w-1$ 个节点，之后再按上面的边长 w 长度均为 1 的解决方法去做。
时间复杂度 $\mathcal{O}(\frac{(m*10)^3}{64} * \log k)$
再细想发现对于每次都将边拆点，其实会造成有很多点的冗余，对于所有目的节点的相同的边，完全可以重复利用拆边得到的点。
因此对于每一个节点 v_1，新增 9 个辅助节点 $v_1, v_2, \ldots, v_9$，v_i 向 v_{i-1} 连一条长度为 1 的边，对于一条 $u \rightarrow v_1$ 长度为 w 的边，直接 u 向 v_w 连一条长度为 1 的边即可。之后再按上面的边长 w 长度均为 1 的解决方法去做。
时间复杂度为 $\mathcal{O}(\frac{(n*10)^3}{64} * \log k)$.

J: 收徒！

将所有棋子 (x_i, y_i) 按照 x 排序后，问题等价为一个最长不下降子序列问题。
通过树状数组等数据结构优化，或通过二分进行优化即可通过。

K: KMP 的馈赠

注意到题目中的 $\mathrm{len}(\mathrm{border})$ 即为 KMP 算法所求的 $\mathtt{nxt}$ 数组。
由 $nxt_i \to i$ 建树，则两个集合 u 和 v 的合并过程等价于从树上找到 $u \to v$ 的路径。
题意转化为能否找到一条路径，使得点权和为 k。
暴力是 $\mathcal{O}(mn^2 \log n)$ 的，无法接受。
暴力写法是 **dis[u] + dis[v] - 2 * dis[lca(u, v)] + val[lca(u, v)]**，而不是 **dis[u] + dis[v] - dis[lca(u, v)]**，我第一遍暴力也写错了 (
事实上，这是点分治应用的经典问题，直接点分治即可通过。时间复杂度为 $\mathcal{O}(mn \log n)$。
验题队写了 dsu on tree 做法，也放过去了。时间复杂度为 $\mathcal{O}(mn \log^2 n)$。
冷知识：本题命制于 ICPC 网络赛第二场前几天，ICPC2 就出了一道 border 树 (

L: 城堡中的皇后

本题同时为高职组的 C 题。
读完题意直接模拟即可。
GPT-4o 写了一个不到一百行的 py 秒了（
彩蛋 1：本来标题是 Queen in the Rook，谐音 Queen in the rock 致敬传奇摇滚乐队皇后乐队
彩蛋 2：第三组样例的局面 1 中，若此时轮到黑方走棋，则局面 2 为黑方的唯一着法
彩蛋 3：第三组样例的局面 3，是初学者容易掉入的四步杀陷阱（

L: 城堡中的皇后

观测到有的队伍模拟写的有点过长，导致难以调试，此处提供一个思路：利用黑方与白方大小写不同的性质，判断吃子时只需要判断 $\text{islower}(u) \oplus \text{islower}(v)$ 是否为真即可，而不用分黑白写两个函数。

此外，注意到 Q = R + B，逻辑也可以简化一些。

M: 速成之道

考察网络流建图，考虑转化为最小割问题。
将每个知识点 i 拆成 i 和 $i^{'}$，建超级源点 S，超级汇点 T。
源点 S 向 i 连容量为 a_i 的边；
i 向 $i^{'}$ 连容量为 b_i 的边；
当存在依赖关系 $x \to y$，则由 $x^{'}$ 向 y 连容量为 inf 的边；
最后由目标点 $X^{'}$ 向汇点 T 连容量为 inf 的边。
求最小割即为答案。

Contents

A: 穿越时空的数字

按照题意把负数翻过来即可。
也可以直接把负数当字符串处理，不涉及爆 int 了。

B: Hello World?

按题意输出即可。
值得注意的是专科组某支队伍在赛前登录时真的把 `l` 改成了 `1`。

E: 做游戏

直接模拟进栈出栈的过程，对于序列的每一个编号 a_i，从小到大将当前还未入过栈且标号小于等于 a_i 的小朋友入栈，依次对于每个序列编号重复该过程接着判断栈顶的编号是否与出栈编号相同，若相同则顺利出栈，若栈为空或不相同则不为出栈序列。

时间复杂度 $O(n)$

F: 加一加二

求中位数最大是多少，可以想到去二分答案。
去检查答案的时候，$x+y$ 次操作施加在序列的前 $\frac{n+1}{2}$ 大显然最优。
由于 $+2$ 可能会造成损失，所以优先将奇数先 $+1$。
注意实现细节，时间复杂度 $O(n\log V)$，V 是值域范围。

G: 开会

首先，求出会议序列。根据题意，聚集地的城市编号是这棵树上 k 个结点的最近公共祖先的结点编号，通过处理树上 dfn 序，找到 dfn 序最小和 dfn 序最大的结点的最近公共祖先即可。
对于询问来说，可以离线处理。按照询问右端点进行排序，当出现编号 x 时，则 x 的倒数第三次出现的位置到 x 的倒数第二次出现的位置将会不合法，每次动态更新这些不合法标记。
可以用线段树维护，每有不合法区间则进行区间加一，不合法区间转为合法区间则区间减一，所询问区间是否合法可以查询区间最值来解决。
时间复杂度 $O(n\log n)$

H: 再次踏上新的旅程

基础 dfs 题。

直接以 $(1, 1)$ 为起始点做 dfs，当到达 (n, m) 时判断经过了多少障碍即可。

正统做法是二维 dp。

I: 我不会数数

数据范围比较小，可以枚举每一个子串，然后判断其是否包含子序列 `weihai`，可以用一个指针维护匹配到了子序列的哪一位。
时间复杂度 $O(n^2 \times t)$，$t = 6$。
也可以预处理出子序列自动机，求出每一位对应下一个字符出现的最早位置。然后对于从第 i 位开始的子串，其贡献是匹配完整个子序列的位置 j 所产生的 $n - j + 1$。预处理时间复杂度 $O(26 \times n)$，累计答案时间复杂度 $O(6 \times n)$。

J: 异或盒子 2

easy 版本只涉及单点修改，全局异或和。
只需要单点修改时在全局异或和中撤销掉原先的贡献，再加上新的贡献即可。
时间复杂度 $\mathcal{O}(n)$。