# IOI2020 中国国家集训队第一阶段作业 解题报告

成都市第七中学 房励行

## 1 Present for Vitalik the Philatelist

### 1.1 试题来源

Codeforces Round #325 (Div. 1) E

### 1.2 试题大意

给出 $n$ 个数 $a_1, a_2, \ldots, a_n$，求出有多少种方案可以从 $n$ 个数中选出一个子集 $s$ 和一个不在子集 $s$ 中的数 $x$，满足 $\gcd(s) > 1$, $\gcd(s \cup \{x\}) = 1$。答案模质数。

### 1.3 数据范围

$2 \le n \le 5 \times 10^5$

$2 \le a_i \le 10^7$

### 1.4 时空限制

时间限制：5s

空间限制：256MB

### 1.5 算法介绍

首先考虑如何求出满足 $\gcd(s) = 1$ 的子集 $s$ 的个数 $A$。可以使用 $\mu$ 函数进行容斥，设 $f(x)$ 表示所有数都含因子 $x$ 的子集个数，则 $f(x) = 2^{cnt_x} - 1$（特别地，$f(1) = 2^n$），其中 $cnt_x$ 表示是 $x$ 的倍数的数的个数。

因此可以得到：

$$A = \sum_{i=1} \mu(i) f(i)$$

得到 $A$ 后同样可以得到 $\gcd(s) > 1$ 的子集 $s$ 的个数 $B$，即 $B = 2^n - A$。

解决了 $s$ 的问题，现在的问题是如何选出一个 $x$ 使得 $\gcd(s \cup \{x\}) = 1$。考虑枚举一个 $x$，那么 $s$ 的选择需要从 $B$ 中减去两部分：一部分是包含 $x$ 的子集，另一部分是不含 $x$ 但与 $x$ 有公因子的子集。

观察发现包含 $x$ 的子集一定含有 $x$ 的因子，因此只需要从 $B$ 中减去与 $x$ 有公因子的子集。

这一部分仍然可以使用 $\mu$ 函数进行容斥。对于 $x$ 的每个非 1 约数 $d$，再从 $B$ 中减去含这个约数的子集个数，即此部分的答案为：

$$\text{answer of } x = B + \sum_{d|x, d\neq 1} \mu(d) f(d)$$

最后将所有枚举 $x$ 所得到的答案求和，即是最终的答案。

对于求 $cnt_x$，观察发现因 $\mu(x) = 0$ 时 $f(x)$ 的值不产生影响，因此只需要求出不含平方因子的数的 $cnt$ 值。可以考虑枚举 $a_i$，将 $a_i$ 分解出 $k$ 个不同质因子，然后 $2^k$ 枚举不含平方因子的因子。

由于值域有限制，分解质因子可以使用欧拉筛预处理。

总时间复杂度为 $O(V + n2^{\omega(n)})$，其中 $V = \max_{i=1}^{n} a_i$，$\omega(n)$ 为 $\leq n$ 的数中所含最多的不同质因子数。

# 2 Three Circuits

## 2.1 试题来源

AtCoder Grand Contest 032 C

## 2.2 试题大意

给定一张 $N$ 个点 $M$ 条边的简单无向连通图，判断能否将图分为三条回路，使得每条边恰在一个回路中。

## 2.3 数据范围

$1 \leq N, M \leq 10^5$

## 2.4 时空限制

时间限制：2s

空间限制：2048MB

## 2.5 算法介绍

本题需按照点的度数分情况讨论。

**情况一：存在点度数为奇数 → 无解**

因图为无向连通图，因此若能分为三条回路，将三条回路顺次连接可以得到一条回路。而由欧拉图的判定定理无向图存在点度为奇数时，不存在这样一条回路，因此该情况无解。

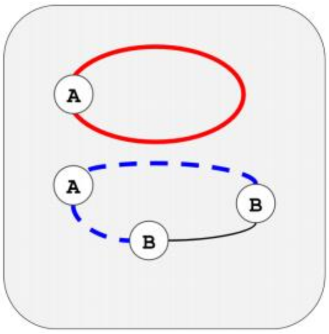
**情况二：点度为偶数，存在点点度 $\geq 6$ → 有解**

因点度均为偶数，可以找出一条欧拉回路。假设存在点 $A$ 度数 $\geq 6$，那么 $A$ 必定被经过至少三次。将这三次经过 $A$ 点之间的路径分别各组成一条回路，即得到三条回路。

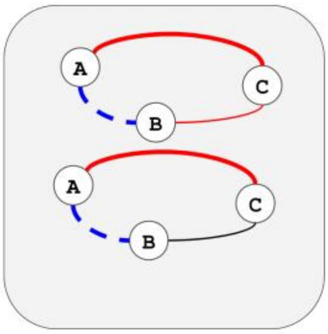
**情况三：点度为偶数，存在至少三个点点度 $= 4$ → 有解**

假设点度 $= 4$ 的三个点分别为 $A$、$B$ 和 $C$，同样，因 $A$ 在欧拉回路中被经过两次，可以将图分为两个包含 $A$ 的环。

若其中一个环中除 $A$ 外也有点被经过了两次，不妨设其为 $B$，如下图所示。那么将该环中 $B$ 至 $B$ 的部分单独取出即可将其再分为两个环，一共三个环。

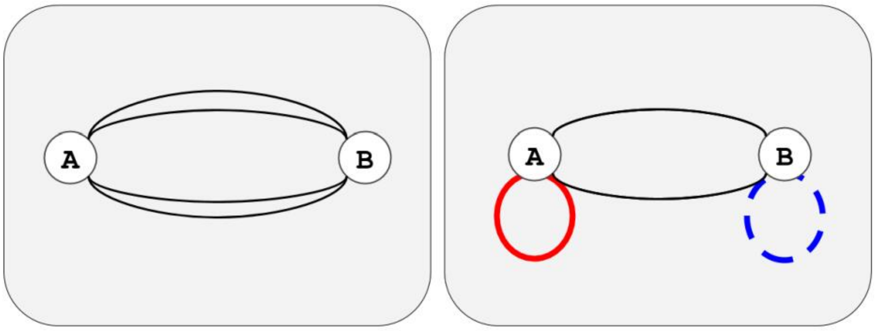



若两个环中的点均只经过一次，因 $B$ 和 $C$ 度数为 4，它们必定在两个环中都出现过，如下图所示。因此，将两个环中的 $A \to B$ 段组成一个环，$B \to C$ 段组成一个环，$C \to A$ 段组成一个环即可得到三个环。




**情况四：点度为偶数，只有两个点点度 $= 4$ → 需再细分讨论**

因只有两个点度数 $=4$，其余的度数均为 2，假设这两个点为 $A$、$B$，那么图有下列两种可能。




左图无法再分为三个环，而右图可以分为 $A \to A$，$A \to B \to A$，$B \to B$ 三个环。

因此在这种情况下，若存在 $A \to A$ 且不经过 $B$ 的环，则问题有解，否则无解。

**情况五：点度为偶数，不多于一个点点度 $=4 \to$ 无解**

当只有一个点点度为 4 时，原图只有两个环，无法拆分出三个环。

当所有点点度均为 2，原图为一个环，无法拆分出三个环。

综合以上所有情况，题目得以解决，时间复杂度为 $O(N+M)$。

# 3 01 on Tree

## 3.1 试题来源

AtCoder Grand Contest 023 F

## 3.2 试题大意

给定一棵 $N$ 个点的有根外向树，每个点有一个权值 $V_i \in \{0,1\}$。现需要求出该外向树的一个拓扑序 $P$，使得序列 $V_{P_1}, V_{P_2}, \ldots, V_{P_N}$ 的逆序对数量最小。

仅需求出这个最小的逆序对数。

## 3.3 数据范围

$1 \le N \le 2 \times 10^5$

## 3.4 时空限制

时间限制：2s

空间限制：256MB

## 3.5 算法介绍

从一个一般化的问题入手：如果每个节点上的权值 $V$ 并非一个数字，而是一个 01 串，不考虑串内部的逆序对，求出拓扑序使得逆序对数最小化。

假设当前第 $i$ 个节点上的 01 串是 $V_i$，其中含有 $C0_i$ 个 0 和 $C1_i$ 个 1。称一个点 $i$ 的“零占比” $R_i = \frac{C0_i}{C1_i}$，其中当 $C1_i = 0$ 时，定义该节点的 $R_i = +\infty$。

直观上来看一个点的 $R_i$ 越大那么这个点就越应该放到前面。可以证明，对于当前树上除根节点外 $R_i$ 最大的点（如果有多个任选一个），一定存在一个使答案最小化的拓扑序，使得该点紧跟在它的父节点之后。下面给出结论的证明：

证明. 令当前 $R$ 最大的非根节点为 $v$，其父节点为 $u$。

假设在拓扑序中 $v$ 之前并非 $u$ 而是 $w$，那么 $w$ 与 $v$ 之间的逆序对数为 $C1_w \times C0_v$。交换 $w$ 与 $v$ 后，$w$ 与 $v$ 间的逆序对数变为 $C1_v \times C0_w$，而除了 $w$ 和 $v$ 间的逆序对数之外，其余逆序对数都不变。因为 $v$ 是 $R$ 最大的节点，有 $R_v \geq R_w$，即 $\frac{C0_v}{C1_v} \geq \frac{C0_w}{C1_w}$，化简得 $C1_v \times C0_w \leq C1_w \times C0_v$，即交换 $w$ 和 $v$ 答案不会变劣。

因此将 $v$ 置于 $u$ 之后答案不会劣于最优答案。 □

至此可以得到一个朴素的算法是，对该树执行 $N-1$ 次下列操作。首先找到 $R$ 最大的非根节点 $v$，然后将它与其父节点 $u$ 合并，并在答案中加上 $u$ 和 $v$ 的贡献 $C1_u \times C0_v$。然而，该算法的时间复杂度是 $O(N^2)$ 的，并不能通过所有测试数据。

为了给该算法加速，可以使用并查集和优先队列。执行每次操作时，从优先队列中取出 $R$ 最大的非根节点，然后合并两个节点时使用并查集进行合并，最后把修改后的 $R$ 值重新放入优先队列。

用此种方法，每一步的时间复杂度均为 $O(\log N)$，因此总时间复杂度为 $O(N \log N)$。