

第八届广西省大学生程序设计大赛 暨 2025 中国—东盟国际大学生程序设计大赛题解

**The 8th Guangxi Collegiate Programming Contest &
The 2025 China-ASEAN International Collegiate Programming Contest Tutorial**

By Colin & His Friends

Hangzhou Dianzi University

May 25, 2025

Overview

Difficulty : Easiest



Hardest

On Site	D	H	A	B	E	K	G	J	L	C	F	I
Idea	D	H	A	K	B	E	J	C	G	L	I	F
Coding	H	D	B	A	E	G	L	K	J	I	C	F
Summary	D	H	A	B	K	G	E	J	C	L	I	F

D. DeepSeek, DeepThink!

输入 "How to xxx?" 输出 "Okay, so I want to figure out how to xxx."

模拟题目要求即可。

- 使用 `getchar` 读入直到 `?`
- 使用 `getline` 读入一整行。

H. Hollow Knight: Silksong

输出给定日期（2017/2/24 与 2025/9/18 之间）到2025年9月18日相差的天数

模拟即可。注意 2020 和 2024 是闰年。

A. Additive Combinatorics

给定整数集 A, B, C , 判断是否 $C = \{x + y : \forall x \in A, y \in B\}$

$O(n^2)$ 枚举所有的组合, 使用 `sort` + `unique` 或 `set` 等工具去重后比较。

复杂度 $O(n^2 \log n)$ 。

B. Beats

给定数列，找到最小的 k ，使得可以将数列划分成若干段：
最后一段的和 $\leq k$ ，其余每段的和恰好 $= k$

注意到这样的 k 一定是某个前缀和。

因此可以 $O(n)$ 枚举 k ，然后用一个数据结构把所有前缀和存下来，快速判断他的每个倍数是否存在，试到不存在的就停。

考虑每个前缀和只会被自己的因数检测到，复杂度 $O(nd(sum))$ 常数很小。

K. Kingdom Rush 5: Alliance

给定两个圆 C_1, C_2 和一个线段 S ，求线段有多少长度被至少一个圆覆盖

先对两个圆分别和 S 所在的直线求交，得到两个线段 S_1, S_2 。

再让两个线段分别和 S 求交，得到 S'_1, S'_2 。

答案即为 S'_1, S'_2 的并的总长度，可以通过单步容斥变成 $|S'_1| + |S'_2| - \text{两者求交的长度}$ 。

这样只需要实现圆和直线求交、线段求交。

E. Elimination

一开始有个字符串 s ，你可以任选一个字符串 t ，每次你可以进行以下两种操作：

- 删除 t 的最后一个字符.
- 若 $s[i \dots j] = t$ ，令 s 变成 $s[i + 1 \dots n]$ ， t 删除最后一个字符.

问最多可以操作多少次操作 2. ($|s| \leq 6000$)

一个串的前缀肯定比它本身更容易匹配到，不会先进行 2 操作再进行 1 操作。

如果答案是 ans ，初始选定的 T 的长度就可以是 ans ，并且是原串的一段区间。

考虑一个较为暴力的做法，设定当前答案为 ans' ，暴力枚举起始点 i ，判定 $S[i, i + ans' + 1)$ 是否可行，可行的话 $ans' \leftarrow ans' + 1$

由于 ans 的最大值是 n ，判定次数至多是 $O(n)$ 的，字符串比较次数是 $O(n^2)$ 的。

可以使用各类字符串算法通过本题。

G. Grand XOR Counting Problem Challenge II

给定数列 a_i, b_i 和参数 k_1, k_2 , 求有多少对 (i, j) 满足 $k_1 \oplus a_i \oplus a_j < k_2 \oplus b_i \oplus b_j$.

两个二进制数 $x < y$, 它们会有一段公共前缀, 然后 x 的下一位为 0, y 的下一位为 1。

对于一个数对 $k_1 \oplus a_i \oplus a_j < k_2 \oplus b_i \oplus b_j$, 我们枚举这段公共前缀的长度, 假设为 c 。

记 $[x]_c$ 为一个二进制数的前 c 位:

$$[k_1 \oplus a_i \oplus a_j]_c = [k_2 \oplus b_i \oplus b_j]_c \Rightarrow [k_1 \oplus a_i \oplus b_i]_c = [k_2 \oplus a_j \oplus b_j]_c$$

把所有数按照 $k_2 \oplus a_j \oplus b_j$ 插入到 Trie 中, 枚举 i , 此时满足要求的 j 为 Trie 的一个子树内的所有点, 此时我们只关心这些二进制数的第 $c+1$ 位。

(需要满足第 $c+1$ 位上, $k_1 \oplus a_i \oplus a_j = 0$, $k_2 \oplus b_i \oplus b_j = 1$)

我们在建立 Trie 时同步记录其子树内第 $c+1$ 位 a_j 和 b_j 取 0/1, 0/1 的各有多少个, 共 4 种情况, 查询时因为已知 a_i, b_i 的值, 可以快速推断出需要的 a_j, b_j 的第 $c+1$ 位需要什么权值, 时间复杂度为 $O(n \log V)$ 。

J. Join the Joint

给定一棵树和两个数列 a_i, b_i , 每次会往 a_i 或 b_i 里加一个数, 求

$$\sum_i \sum_j w_{\text{lca}(a_i, b_i)}$$

特殊限制: $\sum a_i$ 总是不超过 10^6

由特殊限制可以得到不同值的 a_i 最多只有约 1000 个。

可以对每个这样的值, DFS一遍整棵树, 得到这些点与所有其他点的 LCA。

对于每个 a_i 的值, 都存储当前他和所有的 b 能带来的总贡献。

每次加 a 如果是新的值就重新计算贡献, 最多只会做 10^3 次; 否则可以 $O(1)$ 维护。

每次加 b 对所有当前已有的 a 的值更新一遍贡献, 单次只会扫描 10^3 个值。

总复杂度 $O(\sqrt{a_1 + a_2 + \dots}(n + m)) \simeq O(1000(n + m))$

L. Lost in MST

给定一个完全图求最小生成树。 i 和 j 之间的边权为 $\gcd(a_i, a_j) + \text{lcm}(a_i, a_j)$

$$\gcd(u, v) + \text{lcm}(u, v) = \gcd(u, v) + \frac{u \cdot v}{\gcd(u, v)}$$

首先尝试枚举 i 和 $d \mid a_i$, 对于 (i, d) , 确定的 a_j 的值是唯一的, 这样就连出来了所有的可能作为答案的边, 直接跑最小生成树答案就是正确的. 因为 $d \mid \gcd(a_i, a_j)$ 会有多个, 在这种情况下会有重边, 但是这并不影响答案.

$f(d) = d + \frac{u \cdot v}{d}$ 是一个对勾函数, 并且在 $(0, \sqrt{u \cdot v})$ 递减。 $\gcd(u, v) \leq \min(u, v)$ 。

如果直接这样连边, 会有至多 $O(nd(n))$ 条边, 无法通过。

对于数值不同的: 连出来的边数为 $O(n \cdot \log V)$, 对这些边直接跑最小生成树。

对于数值相同的 (假设这个数字是 x , 出现了 cnt 次): 令 x 连出去的最小权值的边为 w , 要么其他的 x 也做出相同行为, 要么内部连边, 那么需要加上 $\min(w, x \times 2) \times (\text{cnt} - 1)$ 。

C. Call for Papers

给定一系列投稿规则与中稿概率，求每篇论文最终被接受的会议影响因子的期望。

注意到会议数并不多 (5000)，因此可以 $O(n^2)$ 求出：

- 每一个会议被拒稿后，按照规则会转投哪个会
- 每一个会议如果是这篇论文第一次投稿，最终被接受的会议影响因子的期望。

注意到答案只与每篇论文第一次投稿的会议有关，因此只需要对每篇论文定位第一次会投稿的会议。

考虑离线+扫描线，`set` 维护当前时刻在接收投稿的会议中影响因子谁最高即可。

总复杂度 $O(n^2 + m \log n)$ 。

F. Faster Elimination

一开始有个字符串 s ，你可以任选一个字符串 t ，每次你可以进行以下两种操作：

- 删除 t 的最后一个字符.
- 若 $s[i \dots j] = t$ ，令 s 变成 $s[i + 1 \dots n]$ ， t 删除最后一个字符.

问最多可以操作多少次操作 2. ($|s| \leq 5 \times 10^5$)

利用观察, 我们可以设计一个 $O(n^2)$ 的简单 dp , 令 F_i 表示 $S[i : n]$ 的答案。

边界: $F_n = 1$, 转移: $F_i \leftarrow \max_{i < j \leq n} \min(F_j, lcp(i, j)) + 1$, 目标是 $\max_{1 \leq i \leq n} F_i$ 。

($lcp(i, j)$ 表示 $S[i : n]$ 和 $S[j : n]$ 的最长公共前缀)

令 pos_i 为 $S[i : n]$ 在反串后缀树上对应的节点, 注意 $lcp(i, j)$ 其实是反串后缀树上 $lca(pos_i, pos_j)$ 对应的长度, 经过这样转化, 可行的 lcp 长度都在 $(root, pos_i)$ 这条路径上。

令 G_t 为后缀树上辅助计算 F 的一个数组。

F. Faster Elimination

那么此时需要支持的操作是:

- 查询 $(root, pos_i)$ 这条后缀树路径上的点 t 中的最大值 G_t
- 对于 $(root, pos_i)$ 这条后缀树路径上的每个点 t 对应的 G_t 对 F_i 取 $\max$, 在进行完操作之后对 len_t 取 $\min$

如果直接使用树链剖分加上 Segment tree beats 的话, 是 $O(n \log^3 n)$ 的。

注意到 len 在树上自上而下是递增的, 那么意味着, 我们需要支持的第二个操作相当于是前缀直接取 len , 其余赋值为 F_i , 同时注意 G_t 变为 len 的次数仅有一次, 通过这种方式我们可以使用树链剖分+线段树做到 $O(n \log^2 n)$ 。

再注意到我们只需要将第一个不被置为 len 的位置对 F_i 取 $\max$ 即可, 因为这个点一定会被统计在内, 可以仅使用倍增/树链剖分在 $O(n \log n)$ 的时间复杂度内解决本题。

I. Identify the Feasibility

定义 $f(x)$ 为有多少对 (i, j) 满足 $a_i + x$ 是 $a_j + x$ 的子集。

求 $\left(\sum_{x=0}^{2^k-1} f(x)\right) \bmod 998244353$

考虑枚举数对 i, j ，计算有多少个 x 满足 $a_i + x$ 是 $a_j + x$ 的子集。

$a_i + x$ 是 $a_j + x$ 的子集，等价与他们的差集 $a_j - a_i$ 也是 $a_j + x$ 的子集。

换言之，只需要找有多少个 x ，满足 $a_j + x$ 是 $a_j - a_i$ 的超集。

考虑 $a_j - a_i \leq 2^k$ ，因此不需要特殊讨论 $a_j + x$ 进位的情况，相当于遍历了 $[0, 2^k - 1]$ 的所有数。因此贡献为 2^{k-d} ，其中 d 为 $a_j - a_i$ 的二进制下 1 的个数。

由于值域较小，可以使用 NTT 计算出所有的 $a_i - a_j$ 。

复杂度 $O(n + 2^k k)$ 。

Thanks for Listening!