第六届广西大学生程序设计大赛 题解

The 6th Guangxi Collegiate Programming Contest Tutorial

By Colin & Eva

Hangzhou Dianzi University

2023 年 6 月 4 日

Overview

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

J. June

Description

输入两个字符串 S, T 。

输出 Good luck to S in T and have fun!

- $1 \leq |S|, |T| \leq 50$。

快乐签到题。

彩蛋：$S =$ highschoolstudents，$T =$ collegeentranceexamination

K. Keyboard

Description

给一个字符串 S_1，每次只能把整个串复制粘贴到自己后面，问是否能变成 S_2 。

- $1 \le |S_1|, |S_2| \le 10^5$

Solution

做法一：直接模拟，直到 $|S_1| \geq |S_2|$ ，然后判断是否 $S_1 = S_2$ 即可。

```
while (s1.length() < s2.length()) s1 = s1 + s1;
if (s1 != s2) puts("Lazy Dog's Great Failure...");
else puts("Smart People's Big Win!");
```

做法二：首先要满足 $|S_2|$ 是 $|S_1|$ 的 **2 的幂次倍**，并且 S_2 是 S_1 循环得到的。

A. Alpha, Beta, Omega

Description

初始牌库中有 n_a 张'Alpha'，n_b 张'Beta'，n_o 张'Omega'。

每次牌堆会随机顺序，然后抽 k 张，然后使用，问最晚第几轮一定抽到'Omega'。

使用'Alpha' 会放入牌堆一张'Beta'，使用'Beta' 会放入牌堆一张'Omega'。

- $0 \le n_a, n_b, n_o \le 3 \times 10^8, 1 \le k \le n_a + n_b + n_o$。

Solution

如果 $k > n_a + n_b$ ，答案就是 1 .

否则答案就是 $\left\lfloor \frac{2 \times n_a + n_b}{k} \right\rfloor + 1$ ，也就是每张牌其实都会发挥最大的作用：

- 如果 $k \le n_a$ ：我们每次都先选尽可能多的'Alpha'，直到不得不选上'Beta' 时，此时'Beta' 一定是足够的，这样剩下的就都是'Beta' 了，一定能执行完。
- 如果 $n_a < k \le n_a + n_b$ ：考虑我们第一次先选全部的'Alpha' 和若干'Beta'，这样剩下的就都是'Beta' 了，一定能执行完。

Bonus：如果有多张牌，每张牌有一个迭代次数 a_i ，被抽中 a_i 次才会变成'Omega'，问第一次抽到'Omega' 的最大轮数。

这个问题存在一个有趣的二分做法，但因为各种原因没有放到本场比赛中。

D. Dream Team

Description

$3n$ 个人，有一些人彼此熟悉，且熟悉关系传递。

现在要分成 n 个 3 人队，每个人的焦虑值是他不熟悉的队友人数，最小化所有人焦虑值的和。

- $1 \leq n \leq 10^6, 1 \leq m \leq 2 \times 10^5$。

Solution

BFS/并查集求出每个连通块，然后对于连通块内，优先组成三人都互相认识的队伍。

接下来每个连通块可能会剩下 $0, 1, 2$ 个人，容易发现优先保留认识的 2 个人在一个队伍里是比较好的。

($2+1$ 的队伍代价是 $1+1+2=4$ ， $1+1+1$ 的队伍代价是 $2+2+2=6$)

于是模拟即可，优先组成 $2+1$ 的组合。

注意如果 1 没有了就需要把一个 2 拆成 $1+1$ ，再继续去组队。

如果 2 没有了就只能组 $1+1+1$ 。

可以证明这样的贪心策略是正确的。复杂度 $O(n)$ 。

B. Brilliant Idea

Description

给一个字符串 S，每次选一个子区间，使得其至少包含两种不同的字符，然后将整个区间修改为两个字符之一。问最多能操作多少次，记作 ans_S 。

对于所有长度为 n 的字符串，问：

1. 有多少个 $ans_S = 0$ ；
2. 有多少个 $ans_S > 0$ 且 ans 有限；
3. 有多少个 $ans_S = \infty$ 。

- $2 \le n \le 10^9$

Solution

- 长度是 2 时，就是样例。
- 长度 ≥ 3 时，可以证明，只要 S 中字符不全相同，就有 $ans_S = \infty$ 。
- abb $\to$ aab $\to$ abb $\to \cdots$
- aab $\to$ abb $\to$ aab $\to \cdots$
- aba $\to$ abb $\to \cdots$
- 因此长度 ≥ 3 时，答案是 $26, 0, 26^n - 26$ ，快速幂计算即可，注意最后的取模。

H. Hide and Seek

Description

计数正整数 x 的个数，首先符合 $x \le a$，且满足下面 m 个方程：

第 j 个方程为 $b_j \bmod x = c_j \ (1 \le j \le m)$

- $1 \le m \le 2 \times 10^5, 1 \le a \le 10^9, 0 \le c_j \le b_j \le 10^9$

Solution

对 $b_j \mod x = c_j$ 化简，等价于 $(b_j - c_j) = kx, k \geq 0$。

每个限制，可以转化成 $x > c_j$，且为 $b_j - c_j$ 的因子。

即 x 为 $g = gcd(b_1 - c_1, b_2 - c_2, \ldots)$ 的因子，且 $x \in (\max c_j, a]$ 。

$O(\sqrt{g})$ 枚举因子，统计满足大小范围要求的个数即为答案。

细节：当 $gcd = 0$ 时，任何区间内的数都合法，答案为 $a - \max c_j$。

E. Evil Capitalist

Description

有 n 个人，每个人初始工资是 a_i 且都不满意。

每次你要选两个不满意的人：

1. 如果他们当前 a_i 相同，两个人都满意。

2. 否则你需要支付给工资较低的那个人（假设为 a_j）$a_i - a_j$ 元，把他的 a_j 变为 a_i，并且他仍然不满意。

求让所有人满意或只有一个人不满意的最小代价。

- $1 \leq n \leq 2 \times 10^5, 1 \leq a_i \leq 10^9$

Solution

如果没有出现同样的数字，那么最大值一直不会消失，答案就是最大值减最小值。

观察发现，如果某个 a_i 出现了偶数次，一定可以把所有不超过 a_i 的都消干净。

(假设有两个 a_i 和某个 $a_j < a_i$ ，可以令 (a_j, a_i) 谈话，最终还是剩下两个 a_i)

那么排序，按照所有出现次数是偶数次的位置进行分段：

每段内让最小的依次滚到最大，代价就是最大值减去最小值。

如果全局最大值也出现了偶数次，那么可以扣掉一个段内的最大差值，否则不行。

比如 1 2 4 5 5 8 9 9 ，分成两段 (1 2 4 5 5) (8 9 9) ，代价为 5-1+9-8-2=3 。

复杂度 $O(n)$ 。

G. GCD Spanning Tree

Description

给定无向连通图 G，询问 k 次。

询问为了保证图中至少存在一个生成树，其所有边的 gcd 为 x 的倍数，最少需要修改原图几条边?

- $1 \leq n, m, k \leq 10^5, 1 \leq w_i, x_i \leq 10^6$

Solution

所有边权的 gcd 为 x 的倍数，等价于所有边权为 x 的倍数。

从倍数考虑，对询问 x，只能使用 x 的倍数的边，如果此时图还不连通，则需要修改联通块个数 -1 条边。

那么每个边最多被其因子的询问访问到，因此枚举量最大为 $O(mw^{1/3})$ 。

对于枚举方式，考虑从 x 角度出发只访问其倍数的边，复杂度是调和级数，即枚举量最大为 $O(w \ln w)$ 。

生成树使用并查集判断，如果每次暴力清空并查集数组会超时。注意到每次并查集改变的元素不超过访问到的边个数，因此记录修改的并查集数组有哪些位置，计算完成后只清空这些修改的即可。

总复杂度 $O(mw^{1/3}\alpha(n) + w \ln w)$ 。

C. Chasing Game

Description

给定一棵树，多次询问：

Eva 从 S 点零时刻出发走到 T 点（唯一简单路径），每秒走一条边。

Colin 从 S' 点零时刻出发，每秒可以不动，或挑一条边走。

问 Colin 最早什么时候能和 Eva 处于同一个点。

- $1 \leq n \leq 2 \times 10^5$

Solution

解法很多，下面是一种讨论比较少的做法：

先假设 Colin 的目的地也是 T 点。

如果 Eva 更早到达 T 点，那么答案一定是 T 点，时间是 S' 到 T 的距离。

如果 Eva 比 Colin 晚或者同时，那么考虑 $S' \to T$ 和 $S \to T$ 重合的一段，Colin 一定可以选择刚到这段的时候转向，转而向 S 走。

观察发现这样的路径恰好是 $S' \to S$ 的路径，因此第二种情况可以认为 Colin 的路径是 $S' \to S$ 。这时是两个人面对面走这段链，比较特殊的是如果路径长度是奇数，那么 Colin 会在距离 Eva 一条边的时候停下 1 秒。

解法过程中需要判断两点距离，以及求某个路径上的第 k 个点，这些都可以通过倍增或树链剖分来实现。复杂度 $O(q\log n)$ 。

I. Indivisible Partition

Description

定义一个字符串是不可分解的，当且仅当他不是某个非平凡前缀重复整数倍得到。

定义一个不可分解的分割是，把一个字符串切成若干段，每段都不可分解。

问 S 有多少种不同的不可分解的分割。

- $1 \leq |S| \leq 5 \times 10^3$

Solution

对 KMP 比较熟悉的话，会知道一个字符串的最小循环节为 $n - border[n]$ ，且其他循环节一定是这个循环节的倍长。

那么一个串不可分解的条件就是 $(n - border[n]) \nmid n$ 或 $border[n] = 0$ 。

在这个基础上进行 dp ，设 $f[i]$ 为前 i 个字符有多少种不同的分割方式。

转移为对于所有的 $j < i$ 且 $S[j+1:i]$ 为不可分解的，$f[i] = \sum f[j]$ 。

对于一个 i ，需要我们判断每个以 i 结尾的子串的 $border$ 。

这可以对 $S[1:i]$ 翻转后求 KMP 得到，总复杂度 $O(n^2)$ 。

空间限制是为了卡掉预处理所有后缀的 KMP 结果的做法。

M. Meteor Shower

Description

在二维平面上给定一个以原点为圆心，r 为半径的圆（地球）。

给 n 个点，每个点有一个移动方向（流星），求判断每一个点延移动方向一直运动（射线）是否会与圆相交。

如果相交，求交点；否则求圆上离射线最近的点。

- $1 \leq n \leq 10^5, -10^9 \leq x, y, dx, dy \leq 10^9$

Solution

比较基础的计算几何题目。

首先取圆心在直线上的投影：

如果起点在这个点的后方（会经过投影点）：

1. 如果投影点在圆内，则会撞，可以构造直角三角形求出交点。
2. 否则不会，把投影点的向量放缩到长度为 r 即为最佳观察点。

如果起点在这个点的前方（不会经过投影点）：

因为输入保证起点不在圆内，所以不会撞，且最佳观察点就是把起点的向量放缩到长度为 r 。

请注意判断是否相撞的过程是没有精度误差的。单次复杂度 $O(1)$ 。

L. Love Choice

Description

有 n 个问题，第 i 个问题有 a_i 个常规选项，有的问题还有一个隐藏选项。

每个题只有一个正确选项（未知），并且如果一个问题有隐藏选项，那么隐藏选项一定是这个问题的正确选项。

每次需要回答完全部问题，才会得知每个题是否做对。

有隐藏选项的题，只有在发现全部常规选项是错误的时候，才会想到有隐藏选项。

每次每道题只会等概率选一个还可能正确的选项回答（已经知道错的必不选）。

问第一次全答对的期望轮数。

- $1 \leq n \leq 5 \times 10^3, 1 \leq a_i \leq 10^9$。

Solution

设随机变量 x_i 为第 i 个题目第一次猜对的轮数。

全部题目第一次猜对的轮数的随机变量 $X = \max_{1\le i\le n}\{x_i\}$ 。

不妨记 $L = \max_{1\le j\le m}\{a_{b_j}+1\}, R = \max(L, \max_{1\le i\le n}\{a_i\})$ ，则 $X \in [L, R]$ 。

X 是恒正的，有 $E(X) = \sum_{k=1}^{\infty} P(X \ge k)$ 。在本题中 $E(x) = L + \sum\limits_{L+1\le k\le R} P(X \ge k)$

而我们知道

$$
\begin{aligned}
P(X \ge k) &= 1 - P(X < k) = 1 - \prod_{1\le i\le n} P(x_i < k) \\
E(X) &= L + \sum_{L+1\le k\le R} \left(1 - \prod_{1\le i\le n} P(x_i < k)\right) \\
&= R - \sum_{L\le k\le R-1} \left(\prod_{1\le i\le n} P(x_i \le k)\right)
\end{aligned}
$$

Solution - Cont'd

对于有隐藏答案的问题，$x_i = a_i + 1 \le L \le k \Rightarrow P(x_i \le k) = 1$ 。

否则我们考察 x_i 的概率分布：

$$\forall 1 \le t \le a_i,\ P(x_i = t) = \frac{a_i - 1}{a_i} \times \frac{a_i - 2}{a_i - 1} \times \cdots \times \frac{a_i - t + 1}{a_i - t + 2} \times \frac{1}{a_i - t + 1} = \frac{1}{a_i}\ 。$$

若 $a_i \ge k$ ，有 $P(x_i \le k) = \frac{k}{a_i}$ ；否则 $P(x_i \le k) = 1$ 。因此我们整理答案：

$$E(x) = R - \sum_{L \le k \le R-1} \prod_{a_i \ge k} \frac{k}{a_i} = R - \sum_{L \le k \le R-1} \left(\prod_{a_i \ge k} \frac{1}{a_i} \right) k^{cnt(k)},\ cnt(k) = \sum_{1 \le i \le n} [a_i \ge k]$$

容易发现 $cnt(k)$ 取值相同的一定是连续的一段值域且幂次前面的系数相同，可以一起算，这是自然数幂和的形式。而 $cnt(k) \le n$ 因此只需要求出 $1 \sim n$ 次幂和的通项公式就可以了。使用拉格朗日插值或其他方法，复杂度 $O(n^2)$ 。

F. Forever Young

Description

给一棵以 1 为根的树，每个点有两个点权 a_i, b_i ，对每个点计算：

$$ans_u = \sum_{x \in subtree_u} \sum_{y \in subtree_u} \min\{|a_x - a_y|, |b_x - b_y|\} \mod 10^9 + 7$$

- $1 \le \sum n \le 5 \times 10^5, 1 \le a_i, b_i \le 10^9$。

Solution

本题中要求 $\min\{|a_i - a_j|, |b_i - b_j|\}$ ，通过 $a + b = \max(a, b) + \min(a, b)$ 转换可得。

考虑把 (a_i, b_i) 二元组视作二维平面上的整点，两个点 (i, j) 之间的产生的贡献实际上等于两个整点之间的切比雪夫距离，即 $\max\{|a_i - a_j|, |b_i - b_j|\}$ 。

通过旋转坐标系可以实现切比雪夫距离转曼哈顿距离：令 $p_i = \frac{a_i+b_i}{2}, q_i = \frac{a_i-b_i}{2}$ ，则 $\max\{|a_i - a_j|, |b_i - b_j|\} = |p_i - p_j| + |q_i - q_j|$。

上式可以通过拆绝对值符号证明。

$$\begin{aligned}\min\{|a_i - a_j|, |b_i - b_j|\} &= |a_i - a_j| + |b_i - b_j| - \max\{|a_i - a_j|, |b_i - b_j|\} \\ &= |a_i - a_j| + |b_i - b_j| - (|p_i - p_j| + |q_i - q_j|)\end{aligned}$$

Solution - Cont'd

接下来只需要解决 4 个形如 $\sum_{i \in \text{subtree } u} \sum_{j \in \text{subtree } u} |a_i - a_j|$ 的问题即可。

注意到，如果我们用可持久化线段树维护出关于 a_i 的有序数组，当我们插入一个新的数字 a_j 的时候，我们只需要知道小于 a_j 的元素的和，以及小于 a_j 的元素的数量，就可以知道 a_j 对新的答案产生的贡献，大于 a_j 的部分同理。

可以通过树上启发式合并的方式，对于每一个点维护出子树的答案。

时间复杂度为 $O(n \log^2 n)$。如果实现细节极其优秀，有可能通过。

Solution - Cont'd

注意到在启发式合并的过程中一些性质被我们浪费了，当我们维护出一个关于 a_i 的有序数组时，答案可以分治的去维护。

假设我们当前维护出的数组为 a ，长度为 n ，取中点 $\mathrm{m} = \frac{n+1}{2}$ 。那么对于 $a[1,n]$ 的答案可以由以下式子表示：

$$\begin{aligned}\text{ans } a[1,n] = &\text{ans } a[1,m] + \text{ans } a[m+1,n] \\ &+ \text{sum } a[m+1,n] \times \text{cnt } a[1,m] \\ &- \text{sum } a[1,m] \times \text{cnt } a[m+1,n]\end{aligned}$$

于是，我们可以通过线段树合并的方式维护出有序数组 a ，在合并的时候按如上方式更新出答案即可。时间复杂度为 $O(n\log n)$ 。

Thank you!