2022 武汉大学新生程序设计竞赛

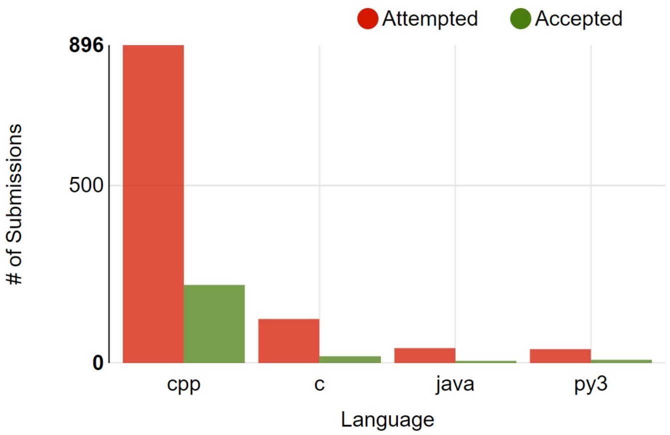
WHU ICPC 集训队

Oct. 15th, 2022

比赛小结

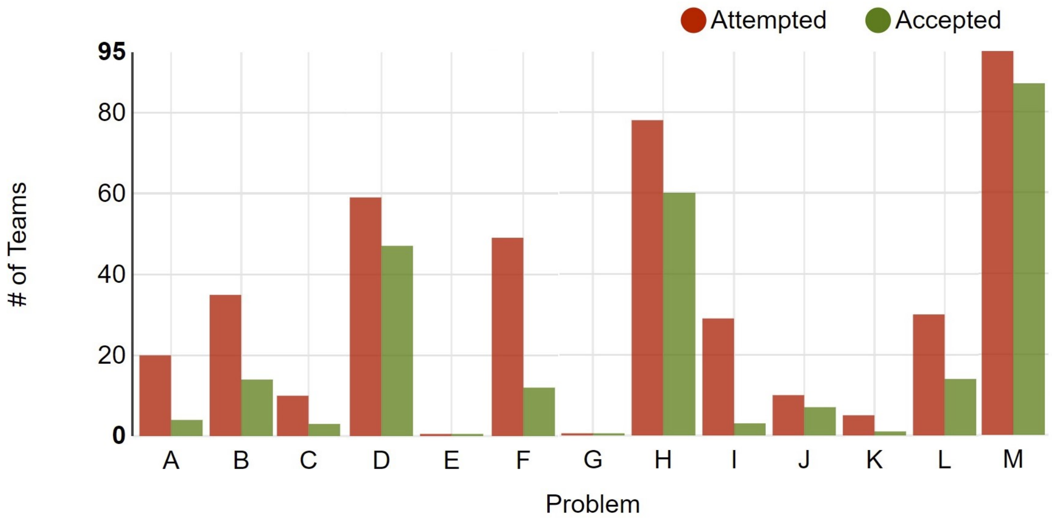
- 本次比赛共收到 1101 份提交代码。
- 其中 254 份代码正确。
- 102 名参赛选手有提交记录。
- 89 名参赛选手至少通过一题。

各题通过情况



Attempted
Accepted
896
500
0
# of Submissions
cpp
c
java
py3
Language

各题通过情况



Attempted
Accepted
95
80
60
40
20
0
# of Teams
A
B
C
D
E
F
G
H
I
J
K
L
M
Problem

A. 蓝线摄影

题意

给一个二维网格，每次能单点修改或者进行一次蓝线拍摄，问最像的一次拍摄有多像。
相关定义见题目。

A. 蓝线摄影

题意

给一个二维网格，每次能单点修改或者进行一次蓝线拍摄，问最像的一次拍摄有多像。
相关定义见题目。

题解

考虑某一次修改对哪些拍摄会产生贡献，发现一次时刻 i 的修改会对区间 $i-m+1$ 之后的拍摄产生影响，即对一个后缀的区间加。
使用差分维护这个区间加的影响即可。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

题解

定义排列的奇偶性为逆序对的奇偶性。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

题解

定义排列的奇偶性为逆序对的奇偶性。

可以发现，任何一次交换后排列的奇偶性必定发生变化，那么使排列有序的交换总次数的奇偶性也是确定的。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

题解

定义排列的奇偶性为逆序对的奇偶性。
可以发现，任何一次交换后排列的奇偶性必定发生变化，那么使排列有序的交换总次数的奇偶性也是确定的。
由于可以反复交换一对位置，所以只要奇交换与偶交换的奇偶性相同就能满足题意。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

题解

定义排列的奇偶性为逆序对的奇偶性。
可以发现，任何一次交换后排列的奇偶性必定发生变化，那么使排列有序的交换总次数的奇偶性也是确定的。
由于可以反复交换一对位置，所以只要奇交换与偶交换的奇偶性相同就能满足题意。
那么如果交换总次数为偶数，两者奇偶性一定相同，否则一定不同。

B. 苹果排序

题意

给一个排列，每次可以交换两个**不同**位置，如果这两个位置差为奇数代价为 1，否则为 -1，问是否能以和为 0 的代价从小到大排序。

题解

定义排列的奇偶性为逆序对的奇偶性。
可以发现，任何一次交换后排列的奇偶性必定发生变化，那么使排列有序的交换总次数的奇偶性也是确定的。
由于可以反复交换一对位置，所以只要奇交换与偶交换的奇偶性相同就能满足题意。
那么如果交换总次数为偶数，两者奇偶性一定相同，否则一定不同。
判断排列奇偶性即可，可以采用直接模拟交换或者求逆序对等方法。

C. 弱连通分量

题意

给一个无向树，现在要把边定向，并最大化所有点中所处的弱连通分量个数的最大值。

C. 弱连通分量

致歉

数据出了一些锅，非常抱歉为大家造成了不好的参赛体验。
另外，广为人知的弱连通分量并不是像题意中所描述的那样。

C. 弱连通分量

致歉

数据出了一些锅，非常抱歉为大家造成了不好的参赛体验。
另外，广为人知的弱连通分量并不是像题意中所描述的那样。

解法

枚举每个点，考虑这个点所处的最多弱连通分量个数。

C. 弱连通分量

致歉

数据出了一些锅，非常抱歉为大家造成了不好的参赛体验。
另外，广为人知的弱连通分量并不是像题意中所描述的那样。

解法

枚举每个点，考虑这个点所处的最多弱连通分量个数。
观察可以发现，弱连通分量实际上是一条极长的单向的有序链，令链的起点与终点为叶子点一定是最优的（这样能匹配出更多的链）。同时，对于给定的根来说，其每个孩子不能独自构成一条经过它的弱连通分量（除非根为起始或终止节点），那么每个孩子所对应的子树的边要么全部指向根要么全部指向叶子。

C. 弱连通分量

解法

那么最终弱连通分量个数即为指向根的叶子和指向叶子的叶子的乘积，对叶子个数进行动态规划，为每个孩子的子树的叶子选择一个方向。对一个度数为 x 的根进行背包的复杂度为 $O(dn)$，则总复杂度为 $O(\sum dn) = O(n^2)$.

C. 弱连通分量

解法

那么最终弱连通分量个数即为指向根的叶子和指向叶子的叶子的乘积，对叶子个数进行动态规划，为每个孩子的子树的叶子选择一个方向。对一个度数为 x 的根进行背包的复杂度为 $O(dn)$，则总复杂度为 $O(\sum dn) = O(n^2)$.

注意到需要考虑每个点都指向根的情况（std 最开始漏判了这个情况）。

D. 甩锅

题意

一个人投掷 x 个 a 面骰子，另一个人投掷 y 个 b 面骰子，按总和取胜负，问是否公平。

D. 甩锅

题解

设 $A = \frac{1+a}{2} \times x, B = \frac{1+b}{2} \times y$，可以发现，对于投掷出 $a_0 = a_1 + a_2 + \cdots + a_x < A$ 的结果，投掷出 $a_0' = (a+1-a_1) + (a+1-a_2) + \cdots + (a+1-a_x) = 2 \times A - a_0 > A$ 的概率与其是相等的，那么 A 与 B 实际上是两者的对称轴。

D. 甩锅

题解

设 $A=\frac{1+a}{2}\times x, B=\frac{1+b}{2}\times y$，可以发现，对于投掷出 $a_0=a_1+a_2+\cdots+a_x<A$ 的结果，投掷出 $a_0'=(a+1-a_1)+(a+1-a_2)+\cdots+(a+1-a_x)=2\times A-a_0>A$ 的概率与其是相等的，那么 A 与 B 实际上是两者的对称轴。

假设 $A=B$，那么对于某次投掷结果 $a_0<b_0$，一定有 $2\times A-a_0>2\times B-b_0$，即对于每一个投 a 面骰子的人输掉的结果，都等概率的存在一个其胜利的结果，因此是公平的。

D. 甩锅

题解

设 $A = \frac{1+a}{2} \times x, B = \frac{1+b}{2} \times y$，可以发现，对于投掷出 $a_0 = a_1 + a_2 + \cdots + a_x < A$ 的结果，投掷出 $a_0' = (a+1-a_1) + (a+1-a_2) + \cdots + (a+1-a_x) = 2 \times A - a_0 > A$ 的概率与其是相等的，那么 A 与 B 实际上是两者的对称轴。

假设 $A = B$，那么对于某次投掷结果 $a_0 < b_0$，一定有 $2 \times A - a_0 > 2 \times B - b_0$，即对于每一个投 a 面骰子的人输掉的结果，都等概率的存在一个其胜利的结果，因此是公平的。

若 $A \neq B$，不妨令 $A < B$，那么对于 $a_0 < b_0$ 对称后的结果的胜负是不确定的，而对于 $a_0 > b_0$ 的情况，却有 $2 \times A - a_0 < 2 \times B - b_0$，即胜利情况不会在对称后再胜利，而失败情况却可能在对称后仍然失败，因此是不公平的。

E. FYC 发气球

题意

有若干组在线发气球的信息，包含过题时间与送气球所需时间，问单人送气球时最小的最大等待时间，带插入。

题解

考虑单组二分答案 T 时，有两组气球信息 $(s_i, v_i), (s_j, v_j), s_i \leqslant s_j$，其中 s_i 表示过题时间，v_i 表示送气球代价。

E. FYC 发气球

题解

考虑单组二分答案 T 时，有两组气球信息 $(s_i, v_i), (s_j, v_j), s_i \leqslant s_j$，其中 s_i 表示过题时间，v_i 表示送气球代价。

若送气球顺序为 (i, j)，假设 i 是刚过题即 s_i 时就送出，且 $s_i + v_i > s_j + T - v_j$，即送完 i 后不能在规定时间内再送 j，那么有

$s_i + v_i > s_j + T - v_j \geqslant s_i + T - v_j \Rightarrow v_i + v_j - T > 0$

则 $v_i + v_j - T > 0 \geqslant s_i - s_j \Rightarrow s_j + v_j > s_i + T - v_i$，即如果送气球顺序为 (j, i)，我们仍不能在规定时间内送好两个气球。

E. FYC 发气球

题解

考虑单组二分答案 T 时，有两组气球信息 $(s_i, v_i), (s_j, v_j), s_i \leqslant s_j$，其中 s_i 表示过题时间，v_i 表示送气球代价。

若送气球顺序为 (i, j)，假设 i 是刚过题即 s_i 时就送出，且 $s_i + v_i > s_j + T - v_j$，即送完 i 后不能在规定时间内再送 j，那么有

$s_i + v_i > s_j + T - v_j \geqslant s_i + T - v_j \Rightarrow v_i + v_j - T > 0$

则 $v_i + v_j - T > 0 \geqslant s_i - s_j \Rightarrow s_j + v_j > s_i + T - v_i$，即如果送气球顺序为 (j, i)，我们仍不能在规定时间内送好两个气球。

假设 i 被前面耽搁了在时间 $x(x > s_i)$ 时被送出，同理可以从 $x + v_i > s_j + T - v_j$ 推出 $x + v_j > s_i + T - v_i$，那么仍然是如果顺序 (i, j) 不行，顺序 (j, i) 也不行。

题解

那么送气球顺序一定是按时间进行排序使得 T 最小，我们令 p_i 为开始送第 i 个气球的时间，那么有 $p_i = \max(p_{i-1} + v_{i-1}, s_i)$，此时有 $T = \max(p_i - s_i + v_i)$。

E. FYC 发气球

题解

那么送气球顺序一定是按时间进行排序使得 T 最小，我们令 p_i 为开始送第 i 个气球的时间，那么有 $p_i = \max(p_{i-1} + v_{i-1}, s_i)$，此时有 $T = \max(p_i - s_i + v_i)$。
可以观察发现 p 实际上分成了若干段，每段开始时即为 $p_i = s_i$ 处。
考虑单点插入后，一定会延长某一段 p 的长度，如果此段 p 与下一段 p 相交，那么两段将会合并，因为只有插入，因此合并段次数是有限的，可以暴力维护。
同时，发现每次延长实际上是一段区间加，只需要维护区间加和全局最大值即可。

E. FYC 发气球

题解

那么送气球顺序一定是按时间进行排序使得 T 最小，我们令 p_i 为开始送第 i 个气球的时间，那么有 $p_i = \max(p_{i-1} + v_{i-1}, s_i)$，此时有 $T = \max(p_i - s_i + v_i)$。

可以观察发现 p 实际上分成了若干段，每段开始时即为 $p_i = s_i$ 处。

考虑单点插入后，一定会延长某一段 p 的长度，如果此段 p 与下一段 p 相交，那么两段将会合并，因为只有插入，因此合并段次数是有限的，可以暴力维护。

同时，发现每次延长实际上是一段区间加，只需要维护区间加和全局最大值即可。

也可以使用动态 dp 维护，支持可删，并更好写。

F. 最短公共超串

题意

给定两个字符串 str_1 和 str_2，请返回同时以 str_1 和 str_2 作为子串的最短字符串。

称字符串 A 为另一个字符串 B 的子串，当且仅当将 B 的首部和尾部分别删除若干个字符（数量均可以为零）后得到 A。例如，abc, bcd, cd, abcde 等都是 abcde 的子串，而 ac, bde 不是。

F. 最短公共超串

题意

给定两个字符串 str_1 和 str_2，请返回同时以 str_1 和 str_2 作为子串的最短字符串。

称字符串 A 为另一个字符串 B 的子串，当且仅当将 B 的首部和尾部分别删除若干个字符（数量均可以为零）后得到 A。例如，abc, bcd, cd, abcde 等都是 abcde 的子串，而 ac, bde 不是。

解法

题目要求找到最短的公共超串 S。

既然 S 是公共超串，那么 S 删掉前面或后面或同时删掉前后可以得到 $str1$ 或 $str2$。

F. 最短公共超串

解法

最简单的，公共超串 $S = \text{str1} + \text{str2}$ 或 $S = \text{str2} + \text{str1}$ 是满足条件的，但是不满足最短。(字符串的加号表示连接操作)

那么既然上述两种情况满足条件，在最优答案 S' 中一定是以 str1 或 str2 的一部分作为开头。

F. 最短公共超串

解法

最简单的，公共超串 $S = \text{str1} + \text{str2}$ 或 $S = \text{str2} + \text{str1}$ 是满足条件的，但是不满足最短。(字符串的加号表示连接操作)

那么既然上述两种情况满足条件，在最优答案 S' 中一定是以 str1 或 str2 的一部分作为开头。

分这两种情况讨论

如果是 str1 作为开头，那么 S' 的前一部分一定就是 str1，否则就没有意义。为了让 S' 最短，同时还包含 str2，要使 S' 最短，那么它的结尾一定是 str2。满足了这两个条件后，如何让它比 $\text{str1} + \text{str2}$ 更短，并得到最短的情况就是我们需要考虑的事情了。

如果是 str2 作为开头，同理 S' 的后一部分一定就是 str2，否则存在浪费。

F. 最短公共超串

解法

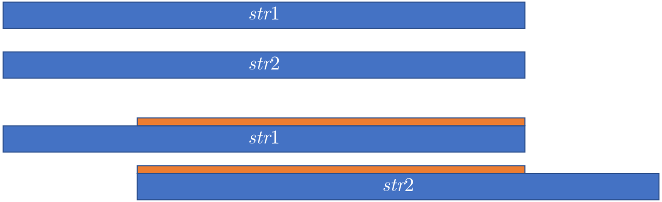
为了使得 S' 最短，就要让 str1 和 str2 的重合部分最多，而判断重合部分可以用哈希来实现。

F. 最短公共超串

解法

为了使得 S' 最短，就要让 str1 和 str2 的重合部分最多，而判断重合部分可以用哈希来实现。

以 str1 + str2 为例，存在一段如图橙色部分（长度也可为 0），此时 `str1[-i:]=str2[:i]`（即 str1 的后 i 个字符与 str2 的前 i 个字符相同），我们要找到最大的 i，就可以尽可能减少 S' 的长度。



F. 最短公共超串

解法

而我们要判断 str1[-i:]=str2[:i]，每次的复杂度是 $O(i)$，而为了找到最大的，要判断 $O(\max(|str1|,|str2|))$ 次，综合复杂度是平方级别的，无法通过。

str1[-i:]=str2[:i]，每次可以按照一个字符的增量，$O(1)$ 判断是否相等。

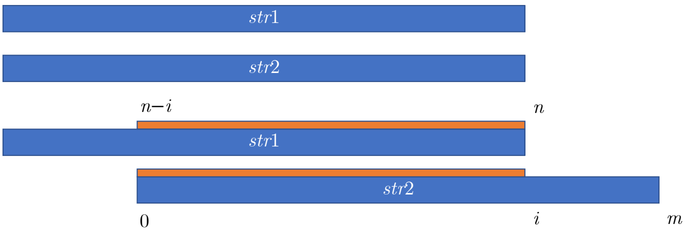
F. 最短公共超串

解法

而我们要判断 `str1[-i:]=str2[:i]`，每次的复杂度是 O(i)，而为了找到最大的，要判断 O(max(|str1|, |str2|)) 次，综合复杂度是平方级别的，无法通过。

`str1[-i:]=str2[:i]`，每次可以按照一个字符的增量，O(1) 判断是否相等。

同时因为上面有两种情况，也需要判断 `str1[-i:]=str2[:i]` 的最大 i。



F. 最短公共超串

总结

此外，当 str1 是 str2 的子串时（反之同理），需要用 KMP 或哈希算法判断是否有这种情况，如果成立，直接输出两者中较长的字符串即可。

因此本题的最佳复杂度是 $O(\max(|str1|, |str2|))$，带 log 的复杂度也可通过。

G. 珂朵莉的帽子

题意

给定一条折线，一个动点沿折线以恒定速度运动，多次询问，每次询问从一个位置以恒定速度运动，最早能与动点相遇的时间。

G. 珂朵莉的帽子

题解

因为所有的 v 都大于 c ，容易得出，如果在时刻 t 可以碰到动点，那么在 t 之后的任意时刻都可以碰到动点。因此可以二分最早碰到动点的时间。

G. 珂朵莉的帽子

题解

因为所有的 v 都大于 c ，容易得出，如果在时刻 t 可以碰到动点，那么在 t 之后的任意时刻都可以碰到动点。因此可以二分最早碰到动点的时间。最优路径显然是直接走直线到相遇点，因为如果走直线可以在 t 之前到达，那么一定存在一种方式使得在 t 时刻正好到达。

G. 珂朵莉的帽子

题解

因为所有的 v 都大于 c ，容易得出，如果在时刻 t 可以碰到动点，那么在 t 之后的任意时刻都可以碰到动点。因此可以二分最早碰到动点的时间。
最优路径显然是直接走直线到相遇点，因为如果走直线可以在 t 之前到达，那么一定存在一种方式使得在 t 时刻正好到达。
可以先二分相遇点在哪一条线段上（如果能够比动点更早到达右端点，说明能够在这条线段上与动点相遇），再在线段上二分时间，复杂度为 $O(\log n + \log t)$ 。也可以直接二分时间，复杂度为 $O(\log n \log t)$ 。

H. 犹太棋 (Easy)

题意

给定长为 n 的棋盘，每次可以下连续的、无棋子的不超过三个位置，问是否先手必胜。

H. 犹太棋 (Easy)

题解

输出”YES”即可通过本题目。

H. 犹太棋 (Easy)

题解

输出” YES” 即可通过本题目。
若棋盘数目是奇数，先手方在 $\frac{1+n}{2}$ 处放一个棋子，将棋盘分成相同长度的两份，对方在一边怎么拿，就在另一边复制他的下法，这样先手方肯定能拿走最后一颗棋子并获胜。
若棋盘数目是偶数，先手方拿走中间两颗即可。
由于本题出题考虑是“让 0 基础选手也能轻松 AC”，没有设置多组数据，无法 hack 掉所有错误做法，抱歉 QwQ.

I. 犹太棋 (Hard)

题意

在上题基础上，有些地方一开始被放了棋子。

题解

本题为 SG 函数的经典模型，SG 函数是用来判断一个公平博弈是否先/后手必胜的工具，值域为非负整数。若一个状态的 SG 值是 0，那么该状态先手必败，否则先手必胜。

而每个状态 SG 值的计算有以下两个定理：

1、一个状态 SG 值等于后继状态 SG 值集合 S 的 mex，其中 $\mathrm{mex}(S) = \min\{x|x \notin S\}$，即最小的，没有出现的非负整数。

2、一个游戏的 SG 值等于子游戏的 SG 值的异或和。

I. 犹太棋 (Hard)

题解

以本题为例，我们令 $f(n)$ 表示一个长为 n 的段（全空）的 SG 值是多少。那么有 $f(n) = \mathtt{mex}(S = \{f(a) \oplus f(b) | a + b = n - 1/2/3\})$，整个的 SG 值等于所有空白段的 SG 值的异或和，即 $ans = \sum_{\oplus} f(n_i)$
复杂度 $O(n^2)$

J. 扔硬币

题意

手上 n 枚硬币，每次可以扔一个去地上或者地上捡一个再扔，最优策略下地上有 x 个正面。$n-x$ 个反面的期望操作次数。

J. 扔硬币

解法

首先显然必须先扔 n 次到地上，考虑如果此时有 i 枚是正面：你会捡起来 $|x-i|$ 枚，并且对于这些硬币，你的策略是：每次拿一枚扔，如果是当前需要的正/反面，就扔下一枚，否则接着扔这一枚，处理这 $|x-i|$ 枚的期望次数是 $2|x-i|$。

J. 扔硬币

解法

首先显然必须先扔 n 次到地上，考虑如果此时有 i 枚是正面：你会捡起来 $|x-i|$ 枚，并且对于这些硬币，你的策略是：每次拿一枚扔，如果是当前需要的正/反面，就扔下一枚，否则接着扔这一枚，处理这 $|x-i|$ 枚的期望次数是 $2|x-i|$。

而扔出 i 个硬币朝上的概率是 $\frac{C_n^i}{2^n}$，于是我们可以得到答案为

$ans = n + \sum_{i=1}^{n} \frac{C_n^i}{2^n} \times 2|x-i|$

利用快速幂和逆元的知识算出 $ans \bmod 998244353$ 即可，复杂度 $O(n \log n)$

K. 序列

题意

已知存在对长度为 n 的 m 次操作，每次操作格式是对 a_i 个不同的位置加 1，求所有情况中，值为 K 的位置最多有多少个，对 $K = 1, 2, 3 \ldots m$ 都输出答案。

K. 序列

解法

首先我们可以发现，只要存在一个长度为 x 的区间的值之和 $f(x)$ 满足 $f(x) = x \times K$，那么 $ans[K] \geqslant x$

于是我们可以从小到大枚举区间 x，考虑这个区间的 $f(x)$ 取值范围 $[L(x), R(x)]$ 是多少。

K. 序列

解法

首先我们可以发现，只要存在一个长度为 x 的区间的值之和 $f(x)$ 满足 $f(x) = x \times K$，那么 $ans[K] \geqslant x$

于是我们可以从小到大枚举区间 x，考虑这个区间的 $f(x)$ 取值范围 $[L(x), R(x)]$ 是多少。

在这个长度为 x 的区间，每个 a_i 能最多放置 $\min(x, a_i)$ 个，最少要放置 $\max(0, a_i - (n - x))$ 个（不然不能保证剩下的 $n - x$ 个位置的放置不重复），所以这 x 个位置的值之和 $f(x)$ 的上下界：

$$R(x) = \sum_{i=1}^{m} \min(x, a_i)$$

$$L(x) = \sum_{i=1}^{m} \max(0, a_i - (n - x))$$

K. 序列

解法

可以发现 $\lceil \frac{L(x)}{x} \rceil \leqslant K \leqslant \lfloor \frac{R(x)}{x} \rfloor$ 这个范围内的 K ，答案可以为 x ，因为 x 是从小到大枚举，我们直接对这些 $ans[K]$ 赋值即可。暴力计算 $L(x), R(x)$ ，并暴力赋值，复杂度是 $O(n^2)$ 的。

考虑怎么优化，观察发现事先对 a_i 排好序，$L(x)$ 和 $R(x)$ 可以用两个单调指针来进行分别维护，而区间赋值的操作可以用线段树实现，此时复杂度为 $O(n\log n)$，可以通过本题。

K. 序列

解法

可以发现 $\lceil \frac{L(x)}{x} \rceil \leqslant K \leqslant \lfloor \frac{R(x)}{x} \rfloor$ 这个范围内的 K，答案可以为 x，因为 x 是从小到大枚举，我们直接对这些 $ans[K]$ 赋值即可。暴力计算 $L(x), R(x)$，并暴力赋值，复杂度是 $O(n^2)$ 的。

考虑怎么优化，观察发现事先对 a_i 排好序，$L(x)$ 和 $R(x)$ 可以用两个单调指针来进行分别维护，而区间赋值的操作可以用线段树实现，此时复杂度为 $O(nlogn)$，可以通过本题。

能否更近一步？再观察，发现赋值操作的区间有单调性，可以看做类似滑动窗口一样的东西，恰好本题 $a_i \leqslant n$，使用桶排序代替快排，可以得到一个 $O(n)$ 线性做法，但实际跑起来提升不是很明显，所以没有对这个做法加以区分。

解法

可以发现 $\lceil \frac{L(x)}{x} \rceil \leqslant K \leqslant \lfloor \frac{R(x)}{x} \rfloor$ 这个范围内的 K，答案可以为 x，因为 x 是从小到大枚举，我们直接对这些 $ans[K]$ 赋值即可。暴力计算 $L(x), R(x)$，并暴力赋值，复杂度是 $O(n^2)$ 的。

考虑怎么优化，观察发现事先对 a_i 排好序，$L(x)$ 和 $R(x)$ 可以用两个单调指针来进行分别维护，而区间赋值的操作可以用线段树实现，此时复杂度为 $O(nlogn)$，可以通过本题。

能否更近一步？再观察，发现赋值操作的区间有单调性，可以看做类似滑动窗口一样的东西，恰好本题 $a_i \leqslant n$，使用桶排序代替快排，可以得到一个 $O(n)$ 线性做法，但实际跑起来提升不是很明显，所以没有对这个做法加以区分。

当然，本题也有一些其他的思考方法与入手点。

L. 最大公约数

题意

给定 n 个数两两的 gcd，还原这 n $(1 \leqslant n \leqslant 1000)$ 个数。

致歉

因为本题造数据纯随机，导致了所有数据新增了“n 个数全部不同”这个新的特质，导致了出现新做法（solution1），对因此导致的不良做题体验表示抱歉。

L. 最大公约数

题意

给定 n 个数两两的 gcd，还原这 n $(1 \leqslant n \leqslant 1000)$ 个数。

致歉

因为本题造数据纯随机，导致了所有数据新增了“n 个数全部不同”这个新的特质，导致了出现新做法（solution1），对因此导致的不良做题体验表示抱歉。

解法

solution1：直接输出出现奇数次的数。

解法

solution2: gcd 有一些性质:

1. $\gcd(x, y) \leqslant \min(x, y)$
2. $\gcd(x, x) = x$

我们找出 gcd 中的最大值，显然他就是 n 个数中的最大值 x_1，然后我们删去 $\gcd(x, x)$，剩余 $n^2 - 1$ 数中最大值肯定是次大值 x_2，但是可能出现 $\gcd(x_1, x_2) \geqslant \gcd(x_3, x_3) = x_3$ ，所以我们还要删去 $\gcd(x_1, x_2)$。
于是我们以此类推的进行，第 i 次找出 n 个数中第 i 大 x_i，并把 x_i 与已找到的 $x_j (1 \leqslant j < i)$ 的 gcd 从初始的 n^2 个数中删去，进行 n 次就还原了原来的数，复杂度 $O(n^2 \log n)$。

M. WHU

题意

给定一个有向无环图，求 S 到 T 路径数目。

M. WHU

解法

根据拓扑序来做即可，具体的：
用 $f(i)$ 表示从 S 出发到达点 i 的路径数，初始 $f(S)=1$，每次选择一个入度为 0 的点 u，对于 u 连向的某点 v，令：$f(v_i){+}{=}f(u)$
然后删去 (u,v) 这条边，直到删去所有边之后，输出 $f(T)$

M. WHU

解法

根据拓扑序来做即可，具体的：
用 $f(i)$ 表示从 S 出发到达点 i 的路径数，初始 $f(S)=1$，每次选择一个入度为 0 的点 u，对于 u 连向的某点 v，令：$f(v_i){+}{=}f(u)$
然后删去 (u,v) 这条边，直到删去所有边之后，输出 $f(T)$
这个过程可以手动模拟，或者写程序来跑，当然因为本题点和边都很少，所以就算暴力硬数问题也不是很大。