2024 (ICPC) Jiangxi Provincial Contest

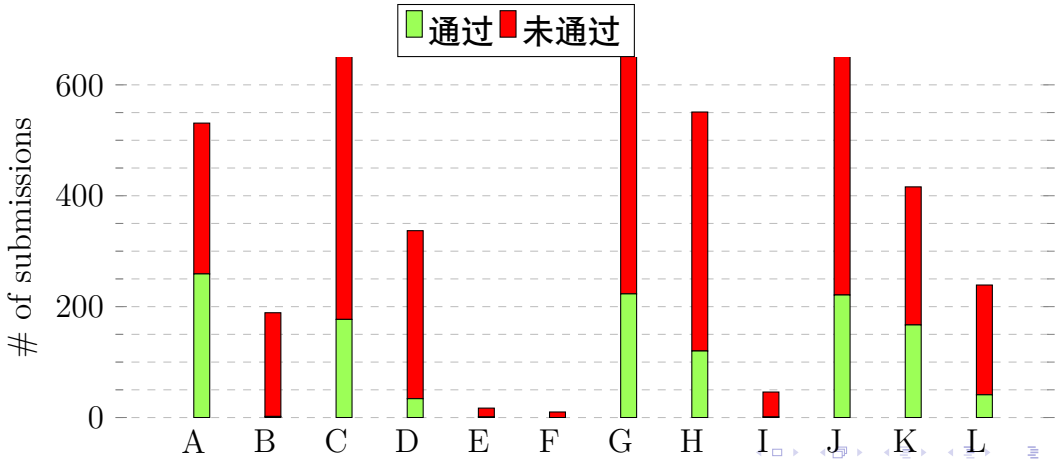
WHU ICPC 命题组

June 2nd, 2024

比赛小结

- 本次比赛共收到 3934 份提交代码。
- 其中 1269 份代码正确。
- 259 个队伍有提交记录。
- 259 个队伍至少通过一题。

各题通过情况



A. Maliang Learning Painting

题意

读入 a, b, c
输出它们的和

题解

输出 $a + b + c$

C. Liar

题意

n 个人每人有一个数，分别宣称自己的数为 $a_1, a_2, \ldots, a_n$。已知所有人的数的和为 s，求至多有多少人没撒谎。

C. Liar

解法

若 $\sum_{i=1}^{n} a_i = s$，则所有人都可以没撒谎，故此时答案为 n。

否则答案一定小于 n。注意到存在一种情况，对于 $i = 1, 2, \ldots, n-1$，第 i 个人没撒谎；而第 n 个人撒谎了，其数实际为 $s - \sum_{i=1}^{n-1} a_i$，仅有第 n 个人撒了谎。故此时答案为 $n-1$。

B. Magic Leeks

题意

初始有长度为 n 的非负整数数组 v_i，表示韭菜高度，星光熠熠在 p 位置。
在每个时刻每个位置的韭菜会长高 k 单位。
每个时刻星光熠熠把当前位置韭菜摘完，然后可以移动到相邻位置。询问某个时刻 t_0 前最多能获得多少韭菜。

B. Magic Leeks

题解

分别考虑新长出来的韭菜和初始韭菜。

注意到当 $t_0 \geq 2 \times n$ 时，初始韭菜一定能吃光，新长韭菜可以先移动到位置 1 或 n 然后等待到还剩余 $n-1$ 秒时移动到另一端，容易证明这样能吃到最多的新长韭菜。

否则注意到转换方向最多只会进行一次，枚举转换方向的位置，使用等差数列和前缀和快速计算新长韭菜和初始韭菜即可。

实现时可能存在一些 corner，需要仔细。

D. Magic LCM

题意

给你一个长 n 的序列 v_i，每次可以选择两个位置，把其中一个位置变为它们的最大公约数，另一个变为最小公倍数，求无限次操作后最大的序列之和。

D. Magic LCM

题解

对于两个数 $x < y$，若它们存在公因子 k，那么有 $\frac{x}{k} + y \times k > y \times (k-1) + y > x + y$，因此操作一定是优的。

考虑操作的本质，对于某个 x 和 y 的共同质因子 p，假设 p^c 和 p^d 分别是这两个数拥有的最大 p 的次幂的因子，那么操作本质上是令最小公倍数拥有 $p^{\max(c,d)}$，最大公约数拥有 $p^{\min(c,d)}$。

那么本题独立考虑每个质因子的次幂，将它们排序，并把更大的次幂放在更右端即可。

实现时使用筛法筛出每个数的最小质因子进行分解，单次分解复杂度是 $\log$ 的。考虑每个数最终最多分解出 7 个质因子，那么最多有 $7n$ 个数可能参与排序，由于非常不满因此排序可以使用 sort，也可以使用桶排序次幂。

E. Boomerang

题意

给一个值域为 $M = 2e7$ 的正整数序列，现在需要找到两个不同的子集，使得它们的和相同。

E. Boomerang

题解

注意到一个长度为 n 的正整数序列的子序列的和最多只有 2^n 种，那么取 $n = 30$，有 $2^n > M \times n$ 存在，因此本题只需要通过任意 30 个数进行判断（实际上存在更紧凑的界，但较为困难，本题不是必须）

对于子集的选取，若有解，一定存在一个子集间互相不交的解，那么 30 个数可以分为三类，属于集合 A 的数，属于集合 B 的数，不在集合中的数。对 30 个数使用 meet in the middle 算法。对于一侧的 3^{15} 种情况，考虑逐个加入数，假设目前准备向有序的长 3^{i-1} 数列加入第 i 个数，实际上是进行了三个有序数列的归并，因此得到有序的 3^{15} 情况的复杂度严格的无 $\log$。类似的，合并两侧的序列继续归并即可。

此处采用手写哈希表同样能够通过。

F. The Ropeways

题意

给定一棵树，树的边长度是 0 或者 1，可以修改，问和一个点 x 距离小于等于 1 的有多少个点。

F. The Ropeways

解法

先考虑静态写法，我们可以很容易的想到把距离为 0 的点集看成一个点，缩点完成后答案就是一个点和周边所有点的点权和。

再考虑只把边长为 1 的边改为边长为 0 的情况，很容易想到可以用并查集来维护点集的所有信息（点集大小，父点集大小，所有子点集大小之和），合并的时候注意子点集大小之和要减去合并上去的对应点集（毕竟也是子点集）。

最后考虑完全动态的情况，我们可以利用线段树分治和可回撤的并查集来实现维护点集信息（就是看成某些边只会在一定时间长度为 0），然后可以在 $O(n \log^2 n)$ 的时间复杂度内解决这道题。

最后就是可以用 LCT 或 ETT 等动态树的方法来维护点集信息，可以在 $O(n \log n)$ 的时间复杂度内解决这道题，由于常数等各种原因，两种算法的实际运行时间并没有相差太大。

G. Multiples of 5

题意

给出一个 11 进制的数，问这个数是否是 5 的倍数。数的长度比较长，以二元组 (x, y) 的形式逐个给出，表示接下来有 x 个 y 将要拼接到右侧。

G. Multiples of 5

题解

考虑到对每个位的数独立考虑对 5 取模的余数，假设第 i 位的数为 x，它们的贡献为 $(11^{i-1} \times x) \bmod 5$。

对连续长度的同一个数的贡献可以等比数列求和，根据实现细节的不同可能需要使用逆元。

注意到 11^i 的末位均为 1，因此其对 5 取模的结果均为 1，那么所有数位的贡献都是相等的，直接考虑有多少个当前数计算即可。

H. Convolution

题意

给定一个二维矩阵 I 以及卷积核 K 的大小，其中卷积核中每个元素只能是 -1 或 0 或 1。矩阵 I 在卷积核的作用下会生成一个新的矩阵 O。
在所有可能的卷积核中，求输出矩阵 O 中的所有元素和的最大值。

H. Convolution

题意

给定一个二维矩阵 I 以及卷积核 K 的大小，其中卷积核中每个元素只能是 -1 或 0 或 1。矩阵 I 在卷积核的作用下会生成一个新的矩阵 O。
在所有可能的卷积核中，求输出矩阵 O 中的所有元素和的最大值。

题解

考虑卷积核 K 中的每个元素产生贡献。K 中的每个元素都对应矩阵 I 中的一个子矩阵。
例如 $K_{i,j}$ 产生的贡献是 $K_{i,j}\times$ 矩阵 $I_{i,j\sim n-k+i,m-l+j}$ 的子矩阵之和，其中的 k,l 是卷积核 K 的长宽。
所以当子矩阵和为负数时，$K_{i,j}$ 取 -1；子矩阵和为正数时，$K_{i,j}$ 取 1。快速求出某个子矩阵和可以用二维前缀和来实现。

I. Neuvillette Circling

题意

给定 n 个点 (x_i, y_i), 对于 $m = 1, 2, ..., \frac{n(n-1)}{2}$ 分别回答：任选圆心，最小化半径，使得至少能完全覆盖 m 条边。
题意等价于：给定 n 个点，对于 $k = 2, ..n$ 分别回答所有的 k 点覆盖圆中最小的半径。

I. Neuvillette Circling

题解

实际上所有的最小覆盖圆，都是由这 n 个点中的两个点（作为直径时）或者三个点确定。所以只需要 $n^2 + n^3$ 分别枚举所有圆，并且再分别计算每个圆覆盖了多少点和它的半径。这样就可以维护所有 k 点覆盖圆中最小的半径。总复杂度为 $O(n^4)$。

J. Gensokyo Autobahn

题意

给出初始的 14 张麻将手牌，判断是否达成国士无双或者七对子。

解法

字符串模拟。
需要注意国士无双是 13 种幺九牌各一张再加上其中任意一张，且牌之间是无序的。
需要注意七对子不能有相同的对子。

K. Party Games

题意

$2 \times n$ 格子从 $(1,1)$ 走四连通进行 dfs，每次随机选择一个没走过的格子扩展。问一共有多少种可能的有标号 dfs 树出现。

K. Party Games

解法

诈骗题。

注意到换行后树实际上截断了，原来一侧的是一条没有选择的链。可以基于这个进行简单的动态规划。

继续观察可以发现，当上次 dfs 时进行了换行后，下一次操作只有一种情况（另一侧延伸是死路），而如果当前没有换行，下一次操作可以选择继续延伸或者不换行。

因为列增加即会导致两种可能，因此方案数为 2^{n-1}。

L. campus

题意

一个 n 个点 m 条边的无向图，每个节点上有 a_i 个人。n 个节点中有 k 个节点为出口，每个出口有一个开放时间 $[l_i, r_i]$。求第 $1 \sim T$ 时刻所有人到最近的开放出口的路径长度之和。

L. campus

解法

首先考虑一个简单版本，所有出口的开放时间无限。
那么就是一个简单的多源最短路的问题，可以建一个超级源点，超级源点到 k 个出口节点连长度为 0 的边。
对于开放时间这个限制，一个朴素的做法就是，对于每个时刻找出对应开放的出口节点来做多源最短路。把 k 个出口节点的开放时间看作时间轴上的 k 条线段，容易发现不同覆盖情况的时间点不超过 $2 \times k$。对这不超过 $2 \times k$ 种情况做多源最短路即可。
时间复杂度:$O(kn \log n)$