

第十七届上海大学程序设计联赛夏季赛

试题分析

蔡盛梁 苗伟华

计算机工程与科学学院

2019 年 7 月 10 日



难度预估

- Very Easy: A,B,I,K
- Easy: E,H
- Medium: F,J
- Medium Hard: C,G
- Hard: D



通过情况

	通过人数	提交人数	首次通过
A	421	437	0:02
B	391	408	0:04
C	0	7	N/A
D	0	147	N/A
E	55	308	0:19
F	15	72	1:14
G	2	6	3:01
H	208	370	0:26
I	155	313	0:18
J	53	178	0:08
K	410	419	0:01



Problem K. 小喵的新朋友

- 最短裁判代码: 145B(c++)
- 最短选手代码: 41B(python3)
- 首次通过: 周晓凡 (0:01)
- 倒过来复读一遍就好了.....



Problem A. 小喵的表达式

- 最短裁判代码：90B(python3)
 - 最短选手代码：92B(python3)
 - 首次通过：罗哲焱 (0:02)
-
- 按题意实现即可。
 - 注意判断除数为 0 的情况。



Problem B. 小喵找东西

- 最短裁判代码: 224B(c++)
 - 最短选手代码: 109B(python3)
 - 首次通过: 李龙旭 (0:04)
-
- C 语言基础练习题, 实现方式有很多。
 - `if (s.substr(0, 4) == "0830") ans++;`



Problem I. 小喵的飞行器

- 最短裁判代码: 451B(c++)
- 最短选手代码: 356B(c++)
- 首次通过: 管文杰 (0:18)

- 两点间距离公式 $d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$
- 计算平方避免使用 pow 函数。



Problem E. 小喵的晚会

- 最短裁判代码: 452B(c++)
 - 最短选手代码: 292B(python3)
 - 首次通过: 周天澜 (0:19)
-
- 一种可行的构造方式是将前 $n - 1$ 个人从小到大排序 (可以使用 `std::sort`), 并把它们依次变为 $1, 2, \dots, n - 1$ 。
 - 请仔细思考如何实现上述过程。
 - 小喵的柠檬数量是 $\sum_{i=1}^n a_i - \sum_{i=1}^{n-1} i$ 。
 - 注意中间计算过程使用 `int` 类型会溢出。



Problem H. 小喵与机器学习

- 最短裁判代码: 602B(c++)
 - 最短选手代码: 698B(python3)
 - 首次通过: 罗哲焱 (0:26)
-
- 按题意实现即可。
 - 第一个正号不需要输出。
 - 大部分 **NO-OUTPUT** 的提交系数全为 0 时忘记输出常数项 0。



Problem F. 小喵看斯诺克

- 最短裁判代码: 1051B(c++)
 - 最短选手代码: 883B(c++)
 - 首次通过: 周天澜 (1:14)
-
- 硬核模拟, 写就完事了。
 - 需要注意的是: 对双方而言剩余的分数是不同的。
 - 一些需要注意的特殊情况:
 - 打丢彩球后可能会造成对方超分。
 - 打进红球后对方实际少了 8 分, 可能打进红球就超分。
 - 最后打平的附加黑球规则。



Problem J. 小喵的门框

- 最短裁判代码：475B(c++)
 - 最短选手代码：356B(c++)
 - 首次通过：周天澜 (0:08)
-
- 分类讨论容易遗漏一些边界情况。
 - 简单搜索即可。
 - 可以考虑每根木棍用新的还是之前剩下的。
 - 可以考虑当前木棍分给几个 a 和 b 使用。
 - 做法似乎非常多，拉过来每个验题人的写法都不一样.....



Problem G. 小喵的超神之路

- 最短裁判代码: 1277B(c++)
 - 最短选手代码: 2050B(c++)
 - 首次通过: 周天澜 (3:01)
-
- 求迷宫中 S 和 A 构成的完全图的最小生成树。
 - 点太多了, 直接对每个点都 BFS 求距离是不行的。
 - 考虑堆优化 Prim 算法的过程, 任选一个点开始 BFS, 每次将最近的点加入集合, 更新最小生成树的权值和。
 - 也可以考虑 Kruskal 算法的过程, 直接进行多源 BFS, 将最短的若干条边加入最小生成树, 并查集判环。
 - 解决此问题需要通过理解算法的过程来实现。



Problem C. 小喵的游戏

最短裁判代码: 861B(c++)

- 先手必败 \iff 路径上所有的边权都出现了偶数次。
- 统计所有边权出现次数为偶数次的路径数量。
- $w_i \rightarrow 2^{w_i}$ (将权值作为二进制位存储)
- DFS 计算前缀链的异或和, 并对每种结果计数。
- 由于数据规模问题, 这个二进制数会非常的长。我们不妨考虑把要异或的那个数对一个质数取模, 即异或 $2^{w_i} \bmod p$ 。这样效果是不变的。想法类似于 (就是) 哈希。
- 事实上, 你只需要让每个数对应一个足够随机的数, 然后判异或就好了。



Problem D. 小喵的函数

● 最短裁判代码: 2385B(c++)

- 如果 k 含有 2, 3, 5, 7 之外的质因数, 答案一定为 0。
- 否则问题变成数位之积中恰好有若干个 2, 3, 5, 7 质因子。
- 直接进行数位 DP 需要五维状态, 内存无法接受。
- 考虑到可能出现的乘积不会很多, 使用 `std::map` 或 `std::unordered_map` 来维护。
- 上述 (验题人) 做法的常数过大, 无法在给定时限内通过。
- 出题人在验题人的代码基础上手写哈希表 (除留余数 + 线性探测) 成功在时限内通过。
- 根据上面的启发, 我们可以预处理出所有满足 $2^a 3^b 5^c 7^d \leq 10^{18}$ 的四元组 (a, b, c, d) , 将四元组通过四维数组与 `int` 做映射。



谢谢~

