

# 第十七届中国地质大学（北京）程序设计大赛

## 网络赛题解

CUGBACM

Carry

2023 年 11 月 23 日



① 写在前面

② 题解

③ 写在最后

① 写在前面

② 题解

③ 写在最后

# 随意说说

- 本文件是文字题解，本次网络赛难度是大致依次从易到难
- 题目所有的 std 文件，即标准程序请见<https://github.com/CarryNotKarry/17thCUGBCPC/tree/main/OnlineContest>

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻



## B. 注意事项

- 模拟，按照题意模拟即可
- 请注意如果一个题没有通过，哪怕错再多次也**不会**对罚时有影响

## CUGBACM

- 按照正方形和长方形的性质即可
- 正方形性质：两组对边分别平行；邻边互相垂直；四条边都相等
- 长方形性质：两组对边分别平行；邻边互相垂直
- 本题的正方形以及长方形的四边都是平行于坐标轴的



## D. 启动!

- 模拟
- 对于每一次读入判断是否名字相同或者金额大于自己的充值金额
- 统计答案即可

# E.Crazy!

- 大致题意：一开始有 50 元，有  $n$  次进行选择：加上  $k$  或者乘以 2
- 假设现在有  $ans$  元，我们每次直接选择  $ans*2$  和  $ans+k$  较大的那个
- 通过数据范围可以计算出需要使用 long long，若使用 C++ 中的 int 则会错误

## F. 双职业竞技场

- 简单思维题
- 显然亵渎作为最后一张牌使用最优
- 对  $a$  数组进行升序排序，设  $b_i$  为使用亵渎前随从  $i$  的血量 ( $1 \leq b_i \leq a_i$ )，为了使亵渎能清场，那么  $b$  数组满足  $b_{i+1} - b_i \leq 1$ 。毒刃的使用数量为  $\sum_{i=1}^n (a_i - b_i)$ ，所以需要最大化  $\sum_{i=1}^n b_i$ 。
- 因为  $b_i \leq a_i$  且  $b_i \leq b_{i-1} + 1$ ，所以我们让  $b_1 = 1$ ， $b_i = \min\{a_i, b_{i-1} + 1\}$  即可。
- 时间复杂度： $\mathcal{O}(n \log n)$
- 空间复杂度： $\mathcal{O}(n)$

## G.kdy 和 lxy 背单词

- 动态规划 (DP)
- 大致题意：给定一个长度为  $n$  的数列，和  $m$  个跳跃。可以打乱  $m$  个跳跃的顺序，问哪一个序列，可以使路线上收集的分数最高

## G.kdy 和 lxy 背单词

- 设  $f[i][j][k][q][p]$  为已经使用了  $i$  个第一种跳跃,  $j$  个第二种跳跃,  $k$  个第三种跳跃,  $q$  个第四种跳跃,  $p$  个第五种跳跃, 可以收集到的最大值。
- 那么这一天的位置就是  $x = i * s[1] + j * s[2] + k * s[3] + q * s[4] + p * s[5]$ 。  $s[i]$  表示第  $i$  种跳跃的跳跃距离。

## G.kdy 和 lxy 背单词

- 

$$f[i][j][k][q][p] = \begin{cases} f[i-1][j][k][q][p] + a[x] & i \geq 1 \\ f[i][j-1][k][q][p] + a[x] & j \geq 1 \\ f[i][j][k-1][q][p] + a[x] & k \geq 1 : \\ f[i][j][k][q-1][p] + a[x] & q \geq 1 \\ f[i][j][k][q][p-1] + a[x] & p \geq 1 \end{cases}$$

- 一共  $m$  天有  $m$  个跳跃，最后一定会把所有的跳跃都使用。所以答案为  $f[t1][t2][t3][t4][t5]$ ， $t[i]$  为每一类的单词数
- 答案会超出 `int`，所有初始化需要为很小，且  $f[0][0][0][0][0]=0$



## H. 爱情的力量是无穷的

- 假设现在蘑菇光和助手都位于点 A，此次他们前往点 N 所选取的路径一定都是各自最少花费可以到达点 N 的方案，也就是同一种方案。故得出，当两人相遇后，之后在前往 N 点的过程中，一定不会分开
- 跑三遍 **Dijkstra** 算法，分别得到点 T, F, N 到每个点所需要的最少乘车数。(这里是 Dijkstra 的抽象，将每一条路径的长度都抽象为 1)。分别存在 `disa`, `disb`, `disc` 三个数组中
- 假设两人是在点  $i$  相遇后一起出发前往点 N，那么此时对应的答案  $ans_i = disa_i * TE + disb_i * FE + disc_i * (TE + FE - S)$
- 之后遍历 N 个点，求出答案  

$$ans = \min\{ans_i \mid i \in \{1, 2, 3, \dots, N\}\}$$



## 1. 谁会赢？

- 大致题意
- 给定一个数字矩阵，两位玩家轮流进行操作，谁先通过操作使得该矩阵变得有序则获胜，问最终的赢家？

# 1. 谁会赢？

- 首先判断该矩阵能否由若干次行列交换变成有序的，如果不能，直接平局。
- 如果先手玩家浅碎经过一步操作即可获胜，则玩家浅碎必胜
- 反之，容易发现剩下的情况中，先手浅碎如果想要取胜，前提必有  $m = 2$ ；后手森柠想要获胜，前提必有  $m = 2$ ，不然游戏将会永远平局

## CUGBACM

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ▶ ↺ 🔍 ↻

## J. 轰轰火花!

- 大致题意
- 实现一种数据结构，实现  $O(\log N)$  内实现数的增删和查询
- 二分、AVL<sup>1</sup>

---

<sup>1</sup>AVL 是一种自平衡二叉搜索树。参考链接：

<https://oi-wiki.org/ds/avl/>

## J. 轰轰火花!

- 正解：采用平衡树存储，避免极端情况每次插入的序列单调使得树退化成链，若为链每次操作时间复杂度变为  $\mathcal{O}(N)$ ，如 Splay
- 其他方法 1：因为该题时间较为宽松、数据较弱，可以使用数组与二分模拟整个过程，例如在插入操作中可以直接对 vector 数组 `a` 进行二分找到位置再插入：  
`a.insert(lower_bound(a.begin(),a.end(),x),x);`
- 其他方法 2：树状数组也可以通过，需要离散化
- 时间复杂度： $\mathcal{O}(n \log n)$
- 空间复杂度： $\mathcal{O}(n)$

## K. 小天天的战争机器人

- 模拟、记忆化搜索
- 大致题意
- 在整个地图上最多有多少个 0 是连在一起的

## K. 小天天的战争机器人

- 使用一个二维数组记录所有已经被访问过的陆地
- 双重 for 循环寻找没有访问过的陆地
- 使用搜索来标记与循环找到的那个陆地连接起来的其他陆地，并使用一个变量来记录下来
- 并查集方法同样可以
- 时间复杂度和空间复杂度均为  $\mathcal{O}(n^2)$

## L. 我很好奇

- 数据结构、线段树
- 大致题意
- 给出一个只含有( 和)的字符串，每次询问  $s_{l_i}, s_{l_i+1}, \dots, s_{r_i}$  的最长合法括号子序列的长度



## L. 我很好奇

- 对每个线段我们维护两个值
- **lu** 未匹配的 ( 数量
- **ru** 未匹配的 ) 数量
- 设线段  $o$  的左右孩子为  $ls, rs$
- 那么线段树中的 `pushup` 操作为

$$t = \min\{lu_{ls}, ru_{rs}\}$$

$$lu_o = lu_{ls} + lu_{rs} - t$$

$$ru_o = ru_{ls} + ru_{rs} - t$$

- 查询操作大同小异，最后答案为区间长度减去为匹配左右括号数量

## M.FFF 团

- 数论
- 大致题意
- 给出  $x$   $y$ , 求  $\gcd(x + k, x + y) \neq 1$  中  $k$  的最小值, 不存在输出-1

## M.FFF 团

- 首先可知，暴力是无法通过的
- 保证  $x < y$
- 根据 gcd 的性质可以得到等式：  

$$\gcd(x+k, y+k) = \gcd(x+k, y+k-(x+k)) = \gcd(x+k, y-x)$$
- 这其中  $y-x$  的值是固定的，我们可以计算  $y-x$  的所有质因子，然后调整  $k$  去匹配即可。若使用根号计算复杂度仍然过高，我们先通过筛法求出  $10^7$  以内的所有素数即可，这样优化之后大约 500 个。
- 时间复杂度： $\mathcal{O}(500n)$
- 空间复杂度： $\mathcal{O}(10^7)$



# Thanks!

祝大家现场赛顺利!