

CSP-J 入门级第二轮复赛模拟赛 • CSP-J 第二轮模拟赛1

本次模拟赛共 4 题，满分 400 分，建议用时 3.5 小时。题目难度依次为 橙 / 黄 / 绿 / 蓝。程序从文件读入、向文件写出，输入输出文件名见下表。

试题概览

题号	题目名称	文件名	时间限制	空间限制	判题方式	数据点个数
T1	技能解锁	skill	1.0 s	128 MB	传统型	20
T2	菜地补水	field	1.0 s	128 MB	传统型	20
T3	集训分组	group	1.0 s	128 MB	传统型	20
T4	共鸣矩阵	matrix	1.0 s	128 MB	传统型	20

编译开关：对于 C++ 语言使用 `-O2 -lm`。**注意事项**

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码请保存在以自己名字拼音命名的文件夹内。
4. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。

目录

- T1 技能解锁
- T2 菜地补水
- T3 集训分组
- T4 共鸣矩阵

T1 技能解锁

时间限制：1 s 内存限制：128 MB

题目描述

小 A 正在玩一款角色扮演游戏。游戏中共有 n 个技能，编号为 $1, \dots, n$ 。

学习技能 i 之前，必须先学会它的若干**前置技能**，这些前置技能的编号都严格小于 i 。一个技能可以没有任何前置技能。

玩家的「能力面板」可用一个长度为 n 的 01 序列 $p_1, p_2, \dots, p_n$ 表示： $p_i = 1$ 表示玩家本身具备施展技能 i 的能力， $p_i = 0$ 表示不具备。

游戏按下述规则结算玩家最终掌握的技能（按编号从小到大依次判定）：

- 若技能 i 的某个前置技能最终**未被掌握**，则技能 i 无法掌握；
- 否则，检查玩家的自身能力：若 $p_i = 1$ ，则技能 i 被掌握；若 $p_i = 0$ ，则不掌握。

一名玩家的**总评分**等于其最终掌握的技能个数。现有 m 名玩家提交了各自的能力面板，请你分别计算每名玩家的总评分。

输入格式

第一行一个整数 n 。

接下来 n 行，第 i 行以整数 d_i 开头，表示技能 i 的前置技能个数；随后是 d_i 个整数，依次为各前置技能的编号。保证这些编号互不相同，且均小于 i 。

接下来一行一个整数 m 。

接下来 m 行，每行 n 个整数 $p_{i,1}, p_{i,2}, \dots, p_{i,n}$ （只可能为 0 或 1），相邻整数之间用空格分隔，描述第 i 名玩家的能力面板。

输出格式

共 m 行，第 i 行输出第 i 名玩家的总评分。

样例 1

输入

```
3
0
1 1
1 2
3
1 1 1
1 0 1
0 0 0
```

输出

```
3
1
0
```

解释

- 第 1 名玩家：能力面板为 1 1 1，三个技能均被掌握，评分为 3。

- 第 2 名玩家：技能 2 的 $p_2 = 0$ 未掌握，技能 3 依赖技能 2 因此也无法掌握，只有技能 1 被掌握，评分为 1。
- 第 3 名玩家：能力面板为 `0 0 0`，评分为 0。

样例 2

输入

```
6
0
1 1
2 1 2
0
1 4
2 3 5
2
1 1 1 1 1 1
1 0 1 1 1 1
```

输出

```
6
3
```

解释

- 第 1 名玩家：六个技能均被掌握，评分为 6。
- 第 2 名玩家：技能 2 未掌握，导致技能 3 与技能 6 均未掌握；被掌握的是技能 1, 4, 5，评分为 3。

数据范围与提示

对于 100% 的数据：

- $1 \leq n, m \leq 1000$;
- $0 \leq d_i < i, 1 \leq a_{i,j} < i$ ，且同一行的前置技能编号互不相同；
- $0 \leq p_{i,j} \leq 1$ 。

提示：请使用 64 位整数存储答案并非必需（答案不超过 n ），但输入量较大，建议使用较快的读入方式。

T2 菜地补水

时间限制：1 s 内存限制：128 MB

题目描述

小 B 有一排 n 块菜地，从左到右依次编号为 $1, 2, \dots, n$ 。第 i 块菜地当前水量与适宜水量相差 a_i ：若 $a_i > 0$ ，表示还缺 a_i 单位水；若 $a_i < 0$ ，表示多出 $|a_i|$ 单位水。

菜地最右端（第 n 块）旁装有一台喷灌机。每次使用时，小 B 需要选择：

- 一个**功率等级** L ($1 \leq L \leq n$) ；
- 一种模式：**补水** 或 **抽水**。

其效果为：

- **补水**模式：第 n 块补 L 单位，第 $n - 1$ 块补 $L - 1$ 单位，第 $n - 2$ 块补 $L - 2$ 单位，.....，第 $n - L + 1$ 块补 1 单位；第 1 到 $n - L$ 块不受影响。
- **抽水**模式：与补水模式影响的田块和数量完全相同，只是把相应的水**抽走**。

也就是说，当功率等级为 L 时，离喷灌机越近的菜地受影响越大，距离每远一块，影响就减少 1 单位，直到不再受影响。

请你计算：最少需要使用喷灌机多少次，才能使所有菜地水量都恰好达到适宜水量（即每块菜地的偏差都变为 0）。

输入格式

第一行一个整数 n 。

第二行 n 个整数 $a_1, a_2, \dots, a_n$ ，相邻整数之间用空格分隔，表示每块菜地水量偏差。

输出格式

输出一行一个整数，表示最少需要使用的次数。

样例 1

输入

```
2
-1 3
```

输出

```
6
```

解释

对第 2 块菜地（也就是喷灌机所在位置）使用抽水模式、功率等级 1，共执行 5 次：每次给第 2 块抽走 1 单位，使 a_2 由 3 变为 -2 。

再使用补水模式、功率等级 2，执行 1 次：第 2 块补 2 单位，第 1 块补 1 单位。

此时两块菜地的偏差分别为 $-1 + 1 = 0$ 、 $-2 + 2 = 0$ ，共使用 6 次。

样例 2

输入

```
5
1 3 -2 -7 5
```

输出

```
26
```

数据范围与提示

对于 100% 的数据：

- $1 \leq n \leq 2 \times 10^5$;
- $-10^{15} \leq a_i \leq 10^{15}$;
- 保证答案不超过 10^9 。

提示：涉及的整数可能超过 32 位，请使用 64 位整数（例如 C/C++ 中的 `long long`）存储。

T3 集训分组

时间限制：1 s 内存限制：128 MB

题目描述

集训队里有 n 名学员排成一排，从左到右依次编号为 $1, 2, \dots, n$ 。第 i 名学员的能力值为 s_i 。

教练打算把学员分成若干个**连续的**小组进行合作训练。每个小组可以包含不超过 K 名学员，且每名学员恰属于一个小组。

由于同组学员会互相交流，训练结束后，小组内每名学员的能力值都会变成该小组中**最高**的能力值。

请你帮助教练安排分组方案，使得训练结束后所有学员能力值之和最大。

输入格式

第一行两个整数 n, K ，相邻整数之间用空格分隔。

第二行 n 个整数 $s_1, s_2, \dots, s_n$ ，表示每名学员的能力值。

输出格式

输出一行一个整数，表示所有学员能力值之和的最大可能值。

样例 1

输入

```
7 3
1 15 7 9 2 5 10
```

输出

```
84
```

解释

最优分组为：1, 2, 3、4、5, 6, 7。

- 第一组最高能力值为 15，组内 3 人能力值均变为 15；
- 第二组仅学员 4，能力值保持 9；
- 第三组最高能力值为 10，组内 3 人能力值均变为 10。

总和为 $15 \times 3 + 9 + 10 \times 3 = 45 + 9 + 30 = 84$ 。注意每个小组的人数可以小于 K 。

样例 2

输入

```
3 5
4 4 4
```

输出

```
12
```

数据范围与提示

对于 100% 的数据：

- $1 \leq n \leq 10^4$;
- $1 \leq K \leq 10^3$;
- $1 \leq s_i \leq 10^5$ 。

提示：答案可能超过 32 位整数的范围，请使用 64 位整数。

T4 共鸣矩阵

时间限制：1 s 内存限制：128 MB

题目描述

有一块 n 行 m 列的共鸣矩阵，共 $n \times m$ 个位置。我们用 (r, c) 表示第 r 行第 c 列的位置。

矩阵满足一条**共鸣规则**：若三个位置

$$(r_1, c_1), \quad (r_1, c_2), \quad (r_2, c_1) \quad (r_1 \neq r_2, c_1 \neq c_2)$$

都被激活，则位置 (r_2, c_2) 会自动激活。

注意：参与共鸣的位置**不会消失**，可以作为其他共鸣的素材反复使用；自动激活的位置同样可以继续参与共鸣。

现在你已拥有 q 个已激活的位置。你可以额外**购买**任意位置使其立即激活。请问：至少还要购买多少个位置，才能最终激活矩阵中的全部 $n \times m$ 个位置？

输入格式

第一行三个整数 n, m, q ，相邻整数之间用空格分隔。

接下来 q 行，每行两个整数 r_i, c_i ，表示已激活的位置。保证给出的位置两两不同。

输出格式

输出一行一个整数，表示最少需要购买的位置数量。

样例 1

输入

```
2 2 3
1 2
2 2
2 1
```

输出

```
0
```

解释

已激活 $(1, 2), (2, 2), (2, 1)$ ，恰好是矩形右上、右下、左下三个角，可直接合成左上角 $(1, 1)$ ，无需购买。

样例 2

输入

```
1 5 3
1 3
1 1
1 5
```

输出

```
2
```

解释

只有一行，无法使用共鸣规则，只能买下缺失的 $(1, 2)$ 与 $(1, 4)$ ，共 2 个。

样例 3

输入

```
4 3 6
1 2
1 3
2 2
2 3
3 1
3 3
```

输出

```
1
```

数据范围与提示

对于 100% 的数据：

- $1 \leq n, m \leq 2 \times 10^5$;
- $0 \leq q \leq \min n \times m, 2 \times 10^5$;
- 保证给出的已激活位置两两不同。

提示： $n \times m$ 可能远大于 q ，不要按矩阵规模开空间；请从「行」与「列」之间的关联入手思考。