

2024 CCF 非专业级软件能力认证

CSP-J/S 2024 第二轮认证

提高级

时间：2024 年 10 月 26 日 14:30 ~ 18:30

题目名称	决斗	超速检测	染色	擂台游戏
题目类型	传统型	传统型	传统型	传统型
目录	duel	detect	color	arena
可执行文件名	duel	detect	color	arena
输入文件名	duel.in	detect.in	color.in	arena.in
输出文件名	duel.out	detect.out	color.out	arena.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	10	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	duel.cpp	detect.cpp	color.cpp	arena.cpp
-----------	----------	------------	-----------	-----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. `main` 函数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. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 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）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
9. 只提供 Linux 格式附加样例文件。
10. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

决斗 (duel)

【题目描述】

今天是小 Q 的生日，他得到了 n 张卡牌作为礼物。这些卡牌属于火爆的“决斗怪兽”，其中，第 i 张卡代表一只攻击力为 r_i ，防御力也为 r_i 的怪兽。

一场游戏分为若干回合。每回合，小 Q 会选择某只怪兽 i 以及另一只怪兽 j ($i \neq j$)，并让怪兽 i 向怪兽 j 发起攻击。此时，若怪兽 i 的攻击力小于等于怪兽 j 的防御力，则无事发生；否则，怪兽 j 的防御被打破，怪兽 j 退出游戏不再参与到剩下的游戏中。一只怪兽在整场游戏中至多只能发起一次攻击。当未退出游戏的怪兽都已发起过攻击时，游戏结束。

小 Q 希望决定一组攻击顺序，使得在游戏结束时，未退出游戏的怪兽数量尽可能少。

【输入格式】

从文件 `duel.in` 中读入数据。

输入的第一行包含一个正整数 n ，表示卡牌的个数。

输入的第二行包含 n 个正整数，其中第 i 个正整数表示第 i 个怪兽的攻击力及防御力 r_i 。

【输出格式】

输出到文件 `duel.out` 中。

输出一行包含一个整数表示游戏结束时未退出游戏的怪兽数量的最小值。

【样例 1 输入】

```
1 5
2 1 2 3 1 2
```

【样例 1 输出】

```
1 2
```

【样例 1 解释】

其中一种最优方案为：第一回合让第 2 只怪兽向第 1 只怪兽发起攻击，第二回合让第 5 只怪兽向第 4 只怪兽发起攻击，第三回合让第 3 只怪兽向第 5 只怪兽发起攻击。此时没有退出游戏的怪兽都进行过攻击，游戏结束。可以证明没有更优的攻击顺序。

【样例 2 输入】

```

1 10
2 136 136 136 2417 136 136 2417 136 136 136

```

【样例 2 输出】

```

1 8

```

【样例 3】

见选手目录下的 *duel/duel3.in* 与 *duel/duel3.ans*。
该样例满足 $\forall 1 \leq i \leq n, r_i \leq 2$ 。

【样例 4】

见选手目录下的 *duel/duel4.in* 与 *duel/duel4.ans*。

【数据范围】

对于所有测试数据，保证： $1 \leq n \leq 10^5$ ， $1 \leq r_i \leq 10^5$ 。

测试点	n	r_i	特殊性质
1 ~ 4	≤ 10	$\leq 10^5$	无特殊性质
5 ~ 10	$\leq 10^5$	≤ 2	
11 ~ 15	≤ 30	$\leq 10^5$	特殊性质 A
16 ~ 20	$\leq 10^5$		无特殊性质

特殊性质 A：保证每个 r_i 在可能的值域中独立均匀随机生成。

超速检测 (detect)

【题目描述】

小 D 新入职了某国的交管部门，他的第一个任务是负责国家的一条长度为 L 的南北主干道的车辆超速检测。为了考考小 D，上司首先需要他解决一个简化的场景。

这个周末，主干道上预计出现 n 辆车，其中第 i 辆车从主干道上距离最南端 d_i 的位置驶入，以 v_i 的初速度和 a_i 的加速度做匀加速运动向北行驶。我们只考虑从南向北的车辆，故 $v_i > 0$ ，但 a_i 可正可负，也可以为零。当车辆行驶到主干道最北端（即距离最南端为 L 的位置）或速度降为 0（这只可能在 $a_i < 0$ 时发生）时，我们认为该车驶离主干道。

主干道上设置了 m 个测速仪，其中第 j 个测速仪位于主干道上距离最南端 p_j 的位置，每个测速仪可以设置开启或关闭。当某辆车经过某个开启的测速仪时，若这辆车的瞬时速度超过了道路限速 V ，那么这辆车就会被判定为超速。注意当车辆驶入与驶出主干道时，如果在对应位置有一个开启的测速仪，这个测速仪也会对这辆车进行测速。

上司首先想知道，如果所有测速仪都是开启的，那么这 n 辆车中会有多少辆车被判定为超速。

其次，为了节能，部门想关闭一部分测速仪。然而，他们不希望漏掉超速的车，也就是说，当 n 辆车里的某辆车在所有测速仪都开启时被判定为超速，他们希望在关闭一部分测速仪以后它依然被判定为超速。上司还想知道在这样的条件下最多可以关闭多少测速仪。

由于 n 很大，上司允许小 D 使用编程解决这两个问题，于是小 D 找到了你。

如果你对于加速度并不熟悉，小 D 贴心地在本题的“提示”部分提供了有关加速度的公式。

【输入格式】

从文件 `detect.in` 中读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数 T ，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

第一行包含四个整数 n, m, L, V ，分别表示车辆数量、测速仪数量、主干道长度和道路限速。

接下来 n 行：

第 i 行包含三个整数 d_i, v_i, a_i 描述一辆车。

最后一行包含 m 个整数 p_1, p_2, \dots, p_m 描述道路上所有测速仪的位置。

【输出格式】

输出到文件 `detect.out` 中。

对于每组数据：输出一行包含两个整数，第一个整数为所有测速仪都开启时被判定为超速的车辆数量，第二个整数为在不漏掉超速车辆的前提下最多可以关闭的测速仪数量。

【样例 1 输入】

```
1 1
2 5 5 15 3
3 0 3 0
4 12 4 0
5 1 1 4
6 5 5 -2
7 6 4 -4
8 2 5 8 9 15
```

【样例 1 输出】

```
1 3 3
```

【样例 1 解释】

在该组测试数据中，主干道长度为 15，限速为 3，在距离最南端 2,5,8,9,15 的位置各设有一个测速仪。

- 第一辆车在最南端驶入，以 3 的速度匀速行驶。这辆车在整个路段上都没有超速。
- 第二辆车在距离最南端 12 的位置驶入，以 4 的速度匀速行驶。在最北端驶离主干道时，它会被距离最南端 15 的测速仪判定为超速。
- 第三辆车在距离最南端 1 的位置驶入，以 1 的初速度、4 的加速度行驶。其在行驶了 $\frac{3^2-1^2}{2 \times 4} = 1$ 的距离，即到达 2 的位置时，速度变为 3，并在之后一直超速。因此这辆车会被除了距离最南端 2 的测速仪以外的其他测速仪判定为超速。
- 第四辆车在距离最南端 5 的位置驶入，以 5 的初速度、-2 的加速度行驶。其在行驶了 $\frac{3^2-5^2}{2 \times (-2)} = 4$ 的距离，即到达 9 的位置时，速度变为 3。因此这辆车在距离最南端 (5,9) 时超速，会被距离最南端 5 和 8 的两个测速仪判定为超速。
- 第五辆车在距离最南端 6 的位置驶入，以 4 的初速度、-4 的加速度行驶。在其行驶了 $\frac{3^2-4^2}{2 \times (-4)} = \frac{7}{8}$ 的距离后，即这辆车到达 $6\frac{7}{8}$ 的位置时，其速度变为 3。因此

这辆车在距离最南端 $[6, 6\frac{7}{8})$ 时超速，但这段区间内没有测速仪，因此不会被判定为超速。

因此第二、三、四辆车会被判定为超速，输出的第一个数为 3。

我们可以关闭距离最南端 2, 8, 9 的三个测速仪，保留 5 和 15 的两个测速仪，此时三辆之前被判定为超速的车依然被判定为超速。可以证明不存在更优方案，因此输出的第二个数为 3。

【样例 2】

见选手目录下的 *detect/detect2.in* 与 *detect/detect2.ans*。

该组样例满足 $n, m \leq 10$ 。

【样例 3】

见选手目录下的 *detect/detect3.in* 与 *detect/detect3.ans*。

该组样例满足特殊性质 A，其中前十组测试数据满足 $n, m \leq 3000$ 。

【样例 4】

见选手目录下的 *detect/detect4.in* 与 *detect/detect4.ans*。

该组样例满足特殊性质 B，其中前十组测试数据满足 $n, m \leq 3000$ 。

【样例 5】

见选手目录下的 *detect/detect5.in* 与 *detect/detect5.ans*。

该组样例满足特殊性质 C，其中前十组测试数据满足 $n, m \leq 3000$ 。

【数据范围】

对于所有测试数据，保证：

- $1 \leq T \leq 20$;
- $1 \leq n, m \leq 10^5$, $1 \leq L \leq 10^6$, $1 \leq V \leq 10^3$;
- $0 \leq d_i < L$, $1 \leq v_i \leq 10^3$, $|a_i| \leq 10^3$;
- $0 \leq p_1 < p_2 < \dots < p_m \leq L$ 。

测试点	$n, m \leq$	特殊性质
1	10	无
2	20	
3	3,000	A
4	10^5	
5	3,000	B
6	10^5	
7	3,000	C
8	10^5	
9	3,000	无
10	10^5	

特殊性质 A: 保证 $a_i = 0$ 。

特殊性质 B: 保证 $a_i > 0$ 。

特殊性质 C: 保证 $a_i < 0$ ，且所有车都不在最北端驶离主干道。

【提示】

与加速度有关的定义和公式如下：

- 匀加速运动 是指物体在运动过程中，加速度保持不变的运动，即每单位时间内速度的变化量是恒定的。
- 当一辆车的初速度为 v_0 、加速度 $a \neq 0$ ，做匀加速运动，则 t 时刻后它的速度 $v_1 = v_0 + a \times t$ ，它的位移（即行驶路程） $s = v_0 \times t + 0.5 \times a \times t^2$ 。
- 当一辆车的初速度为 v_0 、加速度 $a \neq 0$ ，做匀加速运动，则当它的位移（即行驶路程）为 s 时，这辆车的瞬时速度为 $\sqrt{v_0^2 + 2 \times a \times s}$ 。
- 当一辆车的初速度为 v_0 、加速度 $a \neq 0$ ，在它的位移（即行驶路程）为 $\frac{v_1^2 - v_0^2}{2a}$ 时，这辆车的瞬时速度为 v_1 。

如果你使用浮点数进行计算，需要注意潜在的精度问题。

染色 (color)

【题目描述】

给定一个长度为 n 的正整数数组 A ，其中所有数从左至右排成一排。

你需要将 A 中的每个数染成红色或蓝色之一，然后按如下方式计算最终得分：

设 C 为长度为 n 的整数数组，对于 A 中的每个数 A_i ($1 \leq i \leq n$)：

- 如果 A_i 左侧没有与其同色的数，则令 $C_i = 0$ 。
- 否则，记其左侧与其最靠近的同色数为 A_j ，若 $A_i = A_j$ ，则令 $C_i = A_i$ ，否则令 $C_i = 0$ 。

你的最终得分为 C 中所有整数的和，即 $\sum_{i=1}^n C_i$ 。你需要最大化最终得分，请求出最终得分的最大值。

【输入格式】

从文件 `color.in` 中读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数 T ，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

第一行包含一个正整数 n ，表示数组长度。

第二行包含 n 个正整数 A_1, A_2, \dots, A_n ，表示数组 A 中的元素。

【输出格式】

输出到文件 `color.out` 中。

对于每组数据：输出一行包含一个非负整数，表示最终得分的最大可能值。

【样例 1 输入】

```
1 3
2 3
3 1 2 1
4 4
5 1 2 3 4
6 8
7 3 5 2 5 1 2 1 4
```


【样例 1 输出】

```

1 1
2 0
3 8

```

【样例 1 解释】

对于第一组数据，以下为三种可能的染色方案：

1. 将 A_1, A_2 染成红色，将 A_3 染成蓝色 (121)，其得分计算方式如下：

- 对于 A_1 ，由于其左侧没有红色的数，所以 $C_1 = 0$ 。
- 对于 A_2 ，其左侧与其最靠近的红色数为 A_1 。由于 $A_1 \neq A_2$ ，所以 $C_2 = 0$ 。
- 对于 A_3 ，由于其左侧没有蓝色的数，所以 $C_3 = 0$ 。

该方案最终得分为 $C_1 + C_2 + C_3 = 0$ 。

2. 将 A_1, A_2, A_3 全部染成红色 (121)，其得分计算方式如下：

- 对于 A_1 ，由于其左侧没有红色的数，所以 $C_1 = 0$ 。
- 对于 A_2 ，其左侧与其最靠近的红色数为 A_1 。由于 $A_1 \neq A_2$ ，所以 $C_2 = 0$ 。
- 对于 A_3 ，其左侧与其最靠近的红色数为 A_2 。由于 $A_2 \neq A_3$ ，所以 $C_3 = 0$ 。

该方案最终得分为 $C_1 + C_2 + C_3 = 0$ 。

3. 将 A_1, A_3 染成红色，将 A_2 染成蓝色 (121)，其得分计算方式如下：

- 对于 A_1 ，由于其左侧没有红色的数，所以 $C_1 = 0$ 。
- 对于 A_2 ，由于其左侧没有蓝色的数，所以 $C_2 = 0$ 。
- 对于 A_3 ，其左侧与其最靠近的红色数为 A_1 。由于 $A_1 = A_3$ ，所以 $C_3 = A_3 = 1$ 。

该方案最终得分为 $C_1 + C_2 + C_3 = 1$ 。

可以证明，没有染色方案使得最终得分大于 1。

对于第二组数据，可以证明，任何染色方案的最终得分都是 0。

对于第三组数据，一种最优的染色方案为将 A_1, A_2, A_4, A_5, A_7 染为红色，将 A_3, A_6, A_8 染为蓝色 (35152124)，其对应 $C = [0, 0, 0, 5, 0, 1, 2, 0]$ ，最终得分为 8。

【样例 2】

见选手目录下的 *color/color2.in* 与 *color/color2.ans*。

【数据范围】

对于所有测试数据，保证： $1 \leq T \leq 10$ ， $2 \leq n \leq 2 \times 10^5$ ， $1 \leq A_i \leq 10^6$ 。

测试点	n	A_i
1 ~ 4	≤ 15	≤ 15
5 ~ 7	$\leq 10^2$	$\leq 10^2$
8 ~ 10	$\leq 2,000$	$\leq 2,000$
11, 12	$\leq 2 \times 10^4$	$\leq 10^6$
13 ~ 15	$\leq 2 \times 10^5$	≤ 10
16 ~ 20		$\leq 10^6$

擂台游戏 (arena)

【题目描述】

小 S 想要举办一场擂台游戏，如果共有 2^k 名选手参加，那么游戏分为 k 轮进行：

- 第一轮编号为 1, 2 的选手进行一次对局，编号为 3, 4 的选手进行一次对局，以此类推，编号为 $2^k - 1, 2^k$ 的选手进行一次对局。
- 第二轮在只保留第一轮的胜者的前提下，相邻的两位依次进行一场对局。
- 以此类推，第 $k - 1$ 轮在只保留第 $k - 2$ 轮的 4 位胜者的前提下，前两位、后两位分别进行对局，也就是所谓的半决赛。
- 第 k 轮即为半决赛两位胜者的决赛。

确定了游戏晋级的规则后，小 S 将比赛的规则设置为了擂台赛。具体而言，每位选手都有一个能力值 a_1, a_2, \dots, a_{2^k} ，能力值为 $[0, 2^{31} - 1]$ 之内的整数。对于每场比赛，会先抽签决定一个数 0/1，我们将第 R 轮的第 G 场比赛抽到的数记为 $d_{R,G}$ 。抽到 0 则表示编号小的选手为擂主，抽到 1 则表示编号大的选手为擂主。擂主获胜当且仅当他的能力值 $a \geq R$ 。也就是说，游戏的胜负只取决于擂主的能力值与当前比赛是第几轮的大小关系，与另一位的能力值无关。

现在，小 S 先后陆续收到了 n 位选手的报名信息，他们分别告知了小 S 自己的能力值。小 S 会按照报名的先后顺序对选手进行编号为 $1, 2, \dots, n$ 。小 S 关心的是，补充尽量少的选手使总人数为 2 的整次幂，且所有选手进行一次完整的擂台游戏后，所有可能成为总冠军的选手的编号之和为多少。

形式化地，设 k 是最小的非负整数使得 $2^k \geq n$ ，那么应当补充 $(2^k - n)$ 名选手，且补充的选手的能力值可以任取 $[0, 2^{31} - 1]$ 之内的整数。如果补充的选手有可能获胜，也应当计入答案中。

当然小 S 觉得这个问题还是太简单了，所以他给了你 m 个询问 c_1, c_2, \dots, c_m 。小 S 希望你帮忙对于每个 c_i 求出，在只收到前 c_i 位选手的报名信息时，这个问题的答案是多少。

【输入格式】

从文件 `arena.in` 中读入数据。

本题的测试点包含有多组测试数据，但不同测试数据只是通过修改 a_1, a_2, \dots, a_n 得到，其他内容均保持不变，请参考以下格式。其中 \oplus 代表异或运算符， $a \bmod b$ 代表 a 除以 b 的余数。

输入的第一行包含两个正整数 n, m ，表示报名的选手数量和询问的数量。

输入的第二行包含 n 个非负整数 a'_1, a'_2, \dots, a'_n ，这列数将用来计算真正的能力值。

输入的第三行包含 m 个正整数 c_1, c_2, \dots, c_m ，表示询问。

设 K 是使得 $2^K \geq n$ 的最小的非负整数，接下来的 K 行当中，第 R 行包含 2^{K-R} 个数（无空格），其中第 G 个数表示第 R 轮的第 G 场比赛抽签得到的 $d_{R,G} = 0/1$ 。

注意, 由于询问只是将人数凑齐到 $2^k \geq c_i$, 这里的 $k \leq K$, 因此你未必会用到全部的输入值。

接下来一行包含一个正整数 T , 表示有 T 组测试数据。

接下来共 T 行, 每行描述一组数据, 包含 4 个非负整数 X_0, X_1, X_2, X_3 , 该组数据的能力值 $a_i = a'_i \oplus X_{i \bmod 4}$, 其中 $1 \leq i \leq n$ 。

【输出格式】

输出到文件 *arena.out* 中。

共输出 T 行, 对于每组数据, 设 A_i 为第 i ($1 \leq i \leq m$) 组询问的答案, 你只需要输出一行包含一个整数, 表示 $(1 \times A_1) \oplus (2 \times A_2) \oplus \cdots \oplus (m \times A_m)$ 的结果。

【样例 1 输入】

```

1 5 5
2 0 0 0 0 0
3 5 4 1 2 3
4 1001
5 10
6 1
7 4
8 2 1 0 0
9 1 2 1 0
10 0 2 3 1
11 2 2 0 1

```

【样例 1 输出】

```

1 5
2 19
3 7
4 1

```

【样例 1 解释】

共有 $T = 4$ 组数据, 这里只解释第一组。5 名选手的真实能力值为 $[1, 0, 0, 2, 1]$ 。5 组询问分别是对长度为 5, 4, 1, 2, 3 的前缀进行的。

1. 对于长度为 1 的前缀, 由于只有 1 号一个人, 因此答案为 1。

2. 对于长度为 2 的前缀, 由于 2 个人已经是 2 的幂次, 因此不需要进行扩充。根据抽签 $d_{1,1} = 1$ 可知 2 号为擂主, 由于 $a_2 < 1$, 因此 1 号获胜, 答案为 1。
 3. 对于长度为 3 的前缀, 首先 1 号、2 号比赛是 1 号获胜 (因为 $d_{1,1} = 1$, 故 2 号为擂主, $a_2 < 1$), 然后虽然 4 号能力值还不知道, 但 3 号、4 号比赛一定是 4 号获胜 (因为 $d_{1,2} = 0$, 故 3 号为擂主, $a_3 < 1$), 而决赛 1 号、4 号谁获胜都有可能 (因为 $d_{2,1} = 1$, 故 4 号为擂主, 如果 $a_4 < 2$ 则 1 号获胜, $a_4 \geq 2$ 则 4 号获胜)。综上所述, 答案为 $1 + 4 = 5$ 。
 4. 对于长度为 4 的前缀, 我们根据上一条的分析得知, 由于 $a_4 \geq 2$, 所以决赛获胜的是 4 号。
 5. 对于长度为 5 的前缀, 可以证明, 可能获胜的选手包括 4 号、7 号、8 号, 答案为 19。
- 因此, 该组测试数据的答案为 $(1 \times 19) \oplus (2 \times 4) \oplus (3 \times 1) \oplus (4 \times 1) \oplus (5 \times 5) = 5$ 。

【样例 2】

见选手目录下的 *arena/arena2.in* 与 *arena/arena2.ans*。
这组样例满足特殊性质 A。

【样例 3】

见选手目录下的 *arena/arena3.in* 与 *arena/arena3.ans*。
这组样例满足特殊性质 B。

【样例 4】

见选手目录下的 *arena/arena4.in* 与 *arena/arena4.ans*。

【样例 5】

见选手目录下的 *arena/arena5.in* 与 *arena/arena5.ans*。

【数据范围】

对于所有测试数据, 保证: $2 \leq n, m \leq 10^5, 0 \leq a_i, X_j < 2^{31}, 1 \leq c_i \leq n, 1 \leq T \leq 256$ 。

测试点	$T =$	$n, m \leq$	特殊性质 A	特殊性质 B
1 ~ 3	1	8	否	否
4, 5		500	是	
6 ~ 8			否	是
9, 10		5,000		否
11, 12		10^5	是	
13 ~ 15			否	是
16, 17	4			否
18, 19	16			
20, 21	64			
22, 23	128			
24, 25	256			

特殊性质 A: 保证询问的 c_i 均为 2 的幂次。

特殊性质 B: 保证所有的 $d_{R,G} = 0$ 。