

洛谷 CSP-S / NOIP 模拟赛

feecle6418

时间：2023 年 10 月 6 日 14:00 ~ 18:00

题目名称	移动棋子	01 串	松鼠搬家	淘汰赛
题目类型	传统型	传统型	传统型	传统型
可执行文件名	chess	string	squirrel	match
输入文件名	chess.in	string.in	squirrel.in	match.in
输出文件名	chess.out	string.out	squirrel.out	match.out
每个测试点时限	0.5 秒	2.0 秒	3.0 秒	1.5 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB

提交源程序文件名

对于 C++ 语言	chess.cpp	string.cpp	squirrel.cpp	match.cpp
-----------	-----------	------------	--------------	-----------

编译选项

对于 C++ 语言	-lm -O2
-----------	---------

注意事项：

1. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
2. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
7. 上述时间限制以洛谷评测机为准。

移动棋子 (chess)

【题目描述】

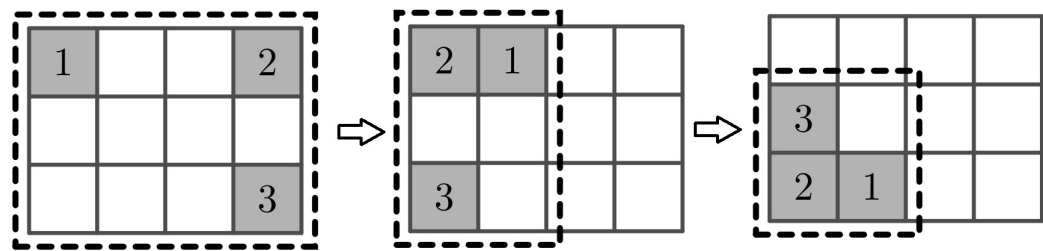
有一个 $H \times W$ 的网格，左上角的格子坐标为 $(1, 1)$ ，右下角的格子坐标为 (H, W) 。网格上有 n 枚棋子，第 i 枚棋子的坐标为 (x_i, y_i) 。你可以无限次（包括 0 次）任意进行以下操作：

- 将所有棋子向上移动一格。
如果某棋子原来在 (x, y) ，则其被移到 $(x - 1, y)$ 。特别地，若原来在 $(1, y)$ ，则其被移到 (H, y) 。
- 将所有棋子向下移动一格。
如果某棋子原来在 (x, y) ，则其被移到 $(x + 1, y)$ 。特别地，若原来在 (H, y) ，则其被移到 $(1, y)$ 。
- 将所有棋子向左移动一格。
如果某棋子原来在 (x, y) ，则其被移到 $(x, y - 1)$ 。特别地，若原来在 $(x, 1)$ ，则其被移到 (x, W) 。
- 将所有棋子向右移动一格。
如果某棋子原来在 (x, y) ，则其被移到 $(x, y + 1)$ 。特别地，若原来在 (x, W) ，则其被移到 $(x, 1)$ 。

定义所有棋子的**最小包围矩形**是满足以下条件的矩形：

- 所有边都平行于网格边缘
- 面积最小
- 所有棋子都在矩形内部

下面是一个例子：



图中灰色的格子为有棋子的格子，坐标分别为 $(1, 1)$, $(1, 4)$, $(3, 4)$ ，虚线即为最小包围矩形。接着，我们依次执行右移操作和上移操作。

你的目标是：在任意多次操作后，让最小包围矩形的面积最小。在此基础上，让操作次数尽可能小。

【输入格式】

从文件 `chess.in` 中读入数据。

第一行三个整数 H, W, n 。

接下来 n 行，第 i 行两个整数 x_i, y_i 。保证棋子的位置不重复。

【输出格式】

输出到文件 *chess.out* 中。

输出两个整数：第一个为任意操作后，最小包围矩形的面积最小值。第二个为达到这个最小值需要的最小步数。

如果你的最小值正确但是步数错误，仍可以得到该测试点 60% 的分数。请注意，即使你不知道最小步数，也需要输出一个符合格式要求的答案！

【样例 1 输入】

```
1 1 10 3
2 1 5
3 1 7
4 1 2
```

【样例 1 输出】

```
1 6 0
```

【样例 1 解释】

不需要任何操作，最小包围矩形即为一个 1×6 的矩形。

【样例 2 输入】

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

【样例 2 输出】

```
1 4 2
```

【样例 2 解释】

该样例即为题目描述中的例子。例子中的方案就是最优方案。

如果你的输出为 4 0，也可以获得 60% 的分数。**注意：如果你只输出一个 4，会因格式错误获得 0 分。**

【样例 3,4】

见下发文件。

【子任务】

本题共 20 个测试点，每个测试点 5 分。所有数据均满足： $1 \leq H, W \leq 10^9, 1 \leq n \leq 10^5$ 。

测试点 1 满足 $n = 1$ 。

测试点 2 ~ 5 满足 $n = 2$ 。

测试点 6, 7 满足 $H = 1, W \leq 500$ 。

测试点 8 ~ 13 满足 $H = 1$ 。

测试点 14, 15 满足 $H, W \leq 50$ 。

测试点 16, 17 满足 $H, W \leq 1000$ 。

测试点 18 满足 $H, W \leq 2 \times 10^5$ 。

测试点 19, 20 无特殊限制。

01 串 (string)

【题目描述】

我们说一个 01 串是好的，当且仅当其能被写为 $(0^k)1(0^k)1 \dots 1(0^k)$ 的形式， k 是非负整数，且中间至少有一个 1。换句话说，其形如：“ k 个 0，一个 1”不停重复，最后以 k 个 0 结尾，且至少有一个 1。

给定一个 01 串，求其最长的好子序列。子序列的定义是删去若干个位置后得到的串。

【输入格式】

从文件 *string.in* 中读入数据。

一行一个 01 串。

【输出格式】

输出到文件 *string.out* 中。

第一行一个整数，表示答案长度。

第二行输出这个最长的好子序列。若有多个输出任意一个均可。

【样例 1 输入】

```
1 0100100000
```

【样例 1 输出】

```
1 7
2 0001000
```

【样例 2 输入】

```
1 0100101
```

【样例 2 输出】

```
1 5
2 01010
```

【样例 3】

见下发文件。

【子任务】

设 len 为 01 串长度。

所有数据均满足 $1 \leq len \leq 5 \times 10^6$ 。保证 01 串中至少有一个 1。

测试点 1 满足 $len \leq 1$ 。

测试点 2, 3, 4 满足 $len \leq 50$ 。

测试点 5, 6, 7 满足 $len \leq 10^4$ 。

测试点 8, 9, 10 满足 $len \leq 10^5$ 且字符串中至多有 50 个 1。

测试点 11, 12, 13 满足 $len \leq 5 \times 10^5$ 。

测试点 14, 15, 16 无特殊限制。

测试点 1 ~ 4 每个测试点 7 分，测试点 5 ~ 10 每个测试点 8 分，测试点 11 ~ 16 每个测试点 4 分。

松鼠搬家 (squirrel)

【题目描述】

有 n 种松鼠，第 i 种松鼠有 b_i 只。初始时，第 i 种松鼠栖息在数轴上坐标为 a_i 的点上，保证 a_i 是整数。

松鼠会经常移动。具体地，有下面三种可能的移动方式：

1. 给定 l, r, x ：种类编号在 $[l, r]$ 的松鼠全部往数轴正方向移动 x 单位。
2. 给定 l, r ：种类编号在 $[l, r]$ 的松鼠全部移动到数轴上同一个 ** 整点 **。由于松鼠很聪明，所以这个整点一定是使得所有松鼠移动距离之和最小的整点。若有多个这样的整点，松鼠会选择坐标最小的。

对于每个 2 操作，你需要输出移动到的整点的坐标。

3. 给定 id, y ，种类为 id 的松鼠坐标不变，但总数变成了 y 只。

形式化地，你要支持对 a, b 进行如下三种操作：

1. 给定 l, r, x ：将 a_l, a_{l+1}, \dots, a_r 加上 x 。
2. 给定 l, r ：设整数 x 满足 $\sum_{l \leq i \leq r} (b_i \times |a_i - x|)$ 是所有整数 x 中最小的（若有多个则取最小的那个），将 a_l, a_{l+1}, \dots, a_r 全部改为 x 。同时，请你输出 x 。
3. 给定 id, y ：将 b_{id} 改为 y 。

【输入格式】

从文件 `squirrel.in` 中读入数据。

第一行两个正整数 n, q 。

接下来一行 n 个整数 $a_1 \sim a_n$ 。

接下来一行 n 个整数 $b_1 \sim b_n$ 。

接下来 q 行，每行第一个整数为 opt 。

1. 若 $opt = 1$ ，再输入三个整数 l, r, x ($1 \leq l \leq r \leq n$) 表示操作 1。
2. 若 $opt = 2$ ，再输入两个整数 l, r ($1 \leq l \leq r \leq n$) 表示操作 2。
3. 若 $opt = 3$ ，再输入两个整数 id, y ($1 \leq id \leq n$) 表示操作 3。

【输出格式】

输出到文件 `squirrel.out` 中。

对于每个 2 操作，按要求输出一行一个整数表示答案。

【样例 1 输入】

```
1 5 8
2 8 1 6 4 9
```

```
3 3 6 4 1 7
4 2 2 4
5 1 1 4 -8
6 2 1 1
7 2 1 3
8 2 4 5
9 1 2 5 6
10 3 3 5
11 2 1 4
```

【样例 1 输出】

```
1 1
2 0
3 -7
4 9
5 -1
```

【样例 2,3】

见下发文件。

【子任务】

本题共 20 个测试点，每个测试点 5 分。所有数据均满足： $1 \leq n \leq 5 \times 10^5$ ， $1 \leq q \leq 5 \times 10^5$ ， $-10^8 \leq a_i, x \leq 10^8$ ， $1 \leq b_i, y \leq 10^8$ 。

测试点编号	$n \leq$	$q \leq$	特殊性质	分值
1 ~ 5	15	15	$ a_i , b_i, x , y \leq 15$	5
6, 7	10^3	10^3	$b_i = y = 1$	5
8, 9	10^3	10^3	无	5
10, 11, 12	10^5	10^5	2 操作不超过 5 次	5
13, 14, 15	2×10^5	2×10^5	无	5
16 ~ 20	5×10^5	5×10^5	无	5

淘汰赛 (match)

【题目描述】

有一场比赛，规则如下：

- 共有 n 人。编号为 i 的人初始有能力值 a_i ， a_i 是一个正整数。所有人按照编号排成一列。
- 你作为裁判，会规划 $n - 1$ 轮对决，每轮对决里你从当前队列里选出一对**相邻的人**，让他们开始决斗。

设对决的两人能力值分别为 p, q 。若 $p \neq q$ ，一定是能力值大的人胜利；否则，你指定哪个人胜利。

胜者留在队列里，败者离开队列。如果队列有空隙，后面的人向前移动把空隙补上。

对决后胜者的能力值增加 1。

- $n - 1$ 轮对决后只剩一个人留在队列里，这个人就是胜者。

不难发现，如果你选出决斗的人的方式不同，你的指定胜利的人不同，最终的胜者就可能不同。我们说第 i 个人“可能获胜”，当且仅当存在一种你的确定赛程以及指定胜者的方式，使得最终的胜者是第 i 个人。

定义函数 $f([a_1, a_2, \dots, a_n]) = [b_1, b_2, \dots, b_n]$ ，其中 a_i 是上面提到的能力值序列， b_i 是 0 或 1，表示第 i 个人是 ($b_i = 1$) 否 ($b_i = 0$) 可能获胜。

你需要解决的问题是：

给定 $n, m, [c_1, c_2, \dots, c_n]$ 和 n 个集合 A_1, \dots, A_n ，满足 $\forall i, \forall j \in A_i, 1 \leq j \leq m$ 。求有多少个正整数序列 $[a_1, a_2, \dots, a_n]$ 满足：

- $\forall i, a_i \in A_i$ 。
- 设 $f(a) = b$ 。如果 $c_i \neq 2$ ，则 $b_i = c_i$ 。

答案对 998244353 取模。

【输入格式】

从文件 `match.in` 中读入数据。

第一行两个正整数 n, m 。

第二行 n 个正整数 c_1, c_2, \dots, c_n 。

接下来 n 行，第 i 行一个长度为 m 的 01 串 s_i ，满足 $s_{i,j} = [j \in A_i]$ 。这里 $[P]$ 当 P 为真时值为 1，否则值为 0。

【输出格式】

输出到文件 `match.out` 中。

输出一行一个正整数，表示答案对 998244353 取模的结果。

【样例 1 输入】

```
1 5 5
2 0 1 2 2 2
3 10100
4 01000
5 01000
6 00010
7 00001
```

【样例 1 输出】

```
1 1
```

【样例 1 解释】

可能的序列 a 有两种: $[1, 2, 2, 4, 5]$ 或 $[3, 2, 2, 4, 5]$ 。

当 $a = [1, 2, 2, 4, 5]$ 时, 考虑第二个人, 能力值为 2。如果我们按照下面方法安排赛程, 就可以让他成为最终胜者:

1. 让加下划线的两人决斗, 红色为胜者: $[1, \underline{2}, 2, 4, 5] \rightarrow [1, \underline{3}, 4, 5]$
2. 让加下划线的两人决斗, 红色为胜者: $[1, \underline{3}, 4, 5] \rightarrow [\underline{4}, 4, 5]$
3. 让加下划线的两人决斗, 红色为胜者: $[\underline{4}, \underline{4}, 5] \rightarrow [\underline{5}, 5]$
4. 让加下划线的两人决斗, 红色为胜者: $[\underline{5}, 5] \rightarrow [\underline{6}]$

最终剩下的人就是初始时的第二个人, 所以第二个人可能获胜。

而第一个人能力值为 1, 无论怎么安排比赛他都不能最终获胜, 所以他不可能获胜。

类似地可以算出 $f([1, 2, 2, 4, 5]) = [0, 1, 1, 1, 1]$, $f([3, 2, 2, 4, 5]) = [1, 1, 1, 1, 1]$, 所以只有 $[1, 2, 2, 4, 5]$ 符合条件。

【样例 2,3,4】

见下发文件。

样例 2 满足测试点 1 的性质。

样例 3 满足测试点 5,6 的性质。

样例 4 满足测试点 8,9,10 的性质。

【子任务】

所有数据均满足 $1 \leq n \leq 40, 1 \leq m \leq 80, 0 \leq c_i \leq 2$ 。

测试点编号	$n \leq$	$m \leq$	特殊性质	分值
1, 2	5	5	无	10
3, 4	30	40	$\prod A_i \leq 10^7$	5
5, 6	30	40	$c_1 = 1, \forall i \neq 1, c_i = 2$	5
7	30	40	存在 $1 \leq i \leq n, c_i = 1, \forall j \neq i, c_j = 2$	24
8, 9, 10	30	40	无	5
11, 12, 13	40	80	无	7