

CCF 全国青少年信息学奥林匹克联赛

NOIP 2025

模拟赛

题目名称	迷途	网格染色	最悪の記者 6	封印
题目类型	传统题	传统题	传统题	传统题
目录	lost	grid	reporter	seal
可执行文件名	lost	grid	reporter	seal
输入文件名	lost.in	grid.in	reporter.in	seal.in
输出文件名	lost.out	grid.out	reporter.out	seal.out
每个测试点时限	500 毫秒	2.0 秒	5.0 秒	2.0 秒
内存限制	512 MiB	256 MiB	512 MiB	1024 MiB
测试点/子任务数目	4	4	4	5
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	lost.cpp	grid.cpp	reporter.cpp	seal.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. C/C++ 中函数 `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. 提交的程序代码文件需要放在与题目英语名称相同的子文件夹中。（建议子文件夹内外各放一份）
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换行）。
5. 选手提交的程序源文件大小不得超过 100 KiB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）或其他可能造成不公平的方法。
8. 因违反上述规定而出现的问题，申诉时一律不予受理。

迷途 (lost)

【题目背景】

小 T 被邪恶的小 S 困在了迷宫中。

【题目描述】

迷宫是一棵深度为 N 的完全二叉树。树上每一个节点都有一个编号：根节点的编号为 1，对于每个编号为 x 的节点，其左儿子的编号为 $2x$ ，右儿子的编号为 $2x+1$ 。小 T 可以从一个节点出发到它的某个儿子去，她要前往位于某一个叶子节点的出口。

幸运的是，小 T 从小 S 那获知了一条从根节点到出口叶子节点的路径，更确切地说，他知道一个正确的长为 $N-1$ 的移动序列，每一项为 L 或 R，小 S 承诺它可以引导小 T 从根节点走到出口处。然而，生性多疑的小 T 怀疑小 S 篡改了路径上一些位置的方向，因此他会在恰好 K 个节点改变策略。策略指顺着走小 S 给出的方向或走恰好相反的方向。小 T 不会在一个节点改变多次策略。

作为旁观者的小 S 想知道，小 T 在他的监控范围 $[A, B]$ 内可能达到的叶子结点的编号之和为多少。答案对 10^9+7 取模。

【输入格式】

从文件 *lost.in* 中读入数据。

第一行两个整数 N, K ；

接下来一行包含一个长为 $N-1$ 的字符串，只包含字母 L 和 R，表示从根到有出口的叶子节点的一条路径；

第三行一个 01 串，为 A 的二进制表示，无前导 0；

第四行一个 01 串，为 B 的二进制表示，无前导 0。

【输出格式】

输出到文件 *lost.out* 中。

输出一个整数 ans ，为所求的对 10^9+7 取模后的值。

【样例 1 输入】

```
1 3 0
2 LR
3 101
4 110
```

【样例 1 输出】

```
1 11
```

【样例 1 解释】

小 T 不会改变他的策略，但是小 S 不知道他在进入根节点之前的策略。因此他可能按照小 S 给出方向先左后右走到节点 5，也可能先右后左走到节点 6，两个叶子节点分别为 A 和 B ，因此答案为 $5 + 6 = 11$ 。

【样例 2 输入】

```
1 4 2
2 LRR
3 1010
4 1110
```

【样例 2 输出】

```
1 37
```

【样例 2 解释】

小 T 经过路径的可能：

- $\{L, L, L\}$;
- $\{L, L, R\}$;
- $\{L, R, L\}$;
- $\{R, L, R\}$;
- $\{R, R, L\}$;
- $\{R, R, R\}$ 。

【样例 3 输入】

```
1 5 2
2 RLLR
3 10010
4 10111
```

【样例 3 输出】

```
1 82
```

【样例 4】

见选手目录下的 *lost/lost4.in* 与 *lost/lost4.ans*。

该样例满足子任务 3 的约束条件。

【样例 5】

见选手目录下的 *lost/lost5.in* 与 *lost/lost5.ans*。

该样例满足子任务 4 的约束条件。

【数据范围】

对于所有数据，

- $0 \leq N \leq 10^3$;
- $0 \leq K \leq N - 1$;
- 保证小 T 可以到达编号为 A 和 B 的两个叶子节点。

子任务编号	$n \leq$	特殊限制	分值
1	10^3	$K = 0$	8
2	25	无	14
3	10^3	A 是 Ena 可能结束的叶子节点最小值, B 是可能的最大值	17
4	10^3	无	61

网格染色 (grid)

【题目背景】

小 S 笑传之 color color board。

【题目描述】

小 S 有一个 N 行 10^9 列的巨大空白网格图。他对网格图进行了 M 次染色，第 i 次染色操作 p_i, l_i, r_i 表示将第 p_i 行的第 $[l_i, r_i]$ 格涂黑。

小 S 认为，如果相邻两行不存在一列都被染黑，则图案差异过于明显，不美观。他决定删去其中的若干行，使得剩下的行构成的网格图满足条件。在此基础上，他希望删除的行数尽可能少。

小 S 想知道，删除的行数最少为多少。同时，他希望得到一个删行最少的方案。

【输入格式】

从文件 *grid.in* 中读入数据。

第一行，两个整数 N, M ；

接下来 M 行中，第 i 行包含三个整数 p_i, l_i, r_i ；

【输出格式】

输出到文件 *grid.out* 中。

第一行输出一个整数 k ，表示最少删除的行数。

第二行输出 k 个整数 a_1, a_2, \dots, a_k ，表示删除行的编号，以任意顺序输出。如果有多解，输出任意一个即可。

【样例 1 输入】

```
1 3 6
2 1 1 1
3 1 7 8
4 2 7 7
5 2 15 15
6 3 1 1
7 3 15 15
```

【样例 1 输出】

```
1 0
```

【样例 1 解释】

第一行与第二行在第七列都被染黑，第二行与第三行在第十五列都被染黑，无需删除任意行即可满足条件。

【样例 2 输入】

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

【样例 2 输出】

```
1 3
2 2 4 5
```

【样例 2 解释】

此样例的网格图如下：

```
01100000.....
00011100.....
00111000.....
00000000.....
10000000.....
```

【样例 3】

见选手目录下的 *grid/grid3.in* 与 *grid/grid3.ans*。

该样例满足子任务 2 的约束条件。

【样例 4】

见选手目录下的 *grid/grid4.in* 与 *grid/grid4.ans*。

该样例满足子任务 3 的约束条件。

【数据范围】

对于所有数据，

- $1 \leq N, M \leq 3 \times 10^5$;

- $1 \leq p_i \leq N, 1 \leq l_i \leq r_i \leq 10^9$ 。

子任务编号	$N, M \leq$	特殊限制	分值
1	20	无	5
2	2000	无	20
3	3×10^5	$\sum_{i=1}^n r_i - l_i + 1 \leq 6 \times 10^5$	30
4	3×10^5	无	45

最悪の記者 6 (reporter)

【题目背景】

记者怎么这么坏。

【题目描述】

小 S 作为校报记者需要报道校优秀团体。具体的，他需要对于所有 1 到 K 之间的整数 i 都报道一个大小恰好为 i 的优秀团体。

现在小 S 收到了候选团体的信息，是一个长度为 N 的序列 a_1, a_2, \dots, a_N ，其中 a_i 表示第 i 个人所属团队编号。但小 S 突然发现可能无法从中选出满足条件的优秀团体，因此他打算选出区间 $[l, r]$ 满足 $1 \leq l \leq r \leq N$ ，并假装由于网络问题只收到了 $[l, r]$ 中的信息。

小 S 想知道，有多少种区间 $[l, r]$ 满足能从中选出符合条件的团体进行报到。

【输入格式】

从文件 *reporter.in* 中读入数据。

第一行，两个正整数 N, K ；

第二行， N 个正整数 a_1, a_2, \dots, a_N 。

【输出格式】

输出到文件 *reporter.out* 中。

输出一行一个整数 ans ，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 6
```

【样例 2 输入】

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

【样例 2 输出】

1 1

【样例 3 输入】

1 6 2
2 5 4 5 2 6 5

【样例 3 输出】

1 5

【样例 4】

见选手目录下的 *reporter/reporter4.in* 与 *reporter/reporter4.ans*。
该样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 *reporter/reporter5.in* 与 *reporter/reporter5.ans*。
该样例满足子任务 3 的约束条件。

【样例 6】

见选手目录下的 *reporter/reporter6.in* 与 *reporter/reporter6.ans*。
该样例满足子任务 4 的约束条件。

【数据范围】

对于所有数据,

- $1 \leq n \leq 10^5$;
- $1 \leq k \leq 4$;
- $1 \leq a_i \leq n$ 。

子任务编号	$n \leq$	特殊限制	分值
1	1000	无	20
2	10^5	A	10
3	10^5	B	30
4	10^5	无	40

- 特殊性质 A: $1 \leq a_i \leq k$;
- 特殊性质 B: $k = 1$ 。

封印 (seal)

【题目背景】

过去已经凝固，我带着回忆向前，只是时常疏于保管，回忆也在改变着各自的形态。这给我的追忆旅程带来些许挑战。

我该在哪里停留？我问我自己。

【题目描述】

小 S 有太多不堪回首的记忆。他需要通过“封印”操作粉饰对过去的回忆。

小 S 的记忆是一个长为 N 的序列 a_1, a_2, \dots, a_N 。定义一次“封印”操作为选择整数 l, r 满足 $1 \leq l < r - 1 < r \leq N$ 且 $a_l = a_r, a_{l+1} = a_{l+2} = \dots = a_{r-1}, a_l \neq a_{l+1}$ ，并将 $a_{l+1}, a_{l+2}, \dots, a_{r-1}$ 都替换为 a_l 。这样一次操作会花费小 S $r - l - 1$ 的精神力。

小 S 的精神力是有限的，因此他希望在操作若干次直到不能再进行“封印”操作的情况下尽可能减少精神力消耗。

小 S 想知道，他所需要花费的最小精神力是多少。

【输入格式】

从文件 *seal.in* 中读入数据。

第一行，一个整数 N 。

第二行， N 个正整数 a_1, a_2, \dots, a_N 。

【输出格式】

输出到文件 *seal.out* 中。

输出一行一个整数 *ans*，表示答案。

【样例 1 输入】

```
1 7
2 1 2 3 2 3 2 1
```

【样例 1 输出】

```
1 7
```

【样例 1 解释】

依次进行 $(l, r) = (3, 5), (2, 6), (1, 7)$ 的操作时， A 会变为 $(1, 2, 3, 2, 3, 2, 1) \rightarrow (1, 2, 3, 3, 3, 2, 1) \rightarrow (1, 2, 2, 2, 2, 2, 1) \rightarrow (1, 1, 1, 1, 1, 1, 1)$ ，此时不存在满足条件的 l, r 。这一系列操作的总代

价为 $1 + 3 + 5 = 9$ 。另一方面, 依次进行 $(l, r) = (2, 4), (4, 6), (1, 7)$ 的操作时, A 会变为 $(1, 2, 3, 2, 3, 2, 1) \rightarrow (1, 2, 2, 2, 3, 2, 1) \rightarrow (1, 2, 2, 2, 2, 2, 1) \rightarrow (1, 1, 1, 1, 1, 1, 1)$, 这一系列操作的总代价为 $1 + 1 + 5 = 7$ 。

【样例 2 输入】

```
1 5
2 1 2 3 4 5
```

【样例 2 输出】

```
1 0
```

【样例 3】

见选手目录下的 *seal/seal3.in* 与 *seal/seal3.ans*。
该样例满足子任务 2 的约束条件。

【样例 4】

见选手目录下的 *seal/seal4.in* 与 *seal/seal4.ans*。
该样例满足子任务 3 的约束条件。

【样例 5】

见选手目录下的 *seal/seal5.in* 与 *seal/seal5.ans*。
该样例满足子任务 5 的约束条件。

【数据范围】

对于所有数据,

- $1 \leq N \leq 2 \times 10^6$;
- $1 \leq a_i \leq N$ 。

子任务编号	$N \leq$	分值
1	20	5
2	500	10
3	5000	25
4	2×10^5	40
5	2×10^6	20