

2025 联赛测试 1

题目名称	数数	画画	打字机	堆
题目类型	传统题	传统题	传统题	传统题
目录	count	paint	typer	heap
可执行文件名	count	paint	typer	heap
输入文件名	count.in	paint.in	typer.in	heap.in
输出文件名	count.out	paint.out	typer.out	heap.out
每个测试点时限	1 秒	3 秒	1 秒	1 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	10	20	4	5
测试点是否等分	是	是	否	否

提交源程序文件名

对于 C++ 语言	count.cpp	paint.cpp	typer.cpp	heap.cpp
-----------	-----------	-----------	-----------	----------

编译选项

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

注意事项（请仔细阅读）

1. 测试机器：CPU(AMD Ryzen 7 5700G 3.80 GHz), RAM 32G。
2. 测试环境：Windows 10, g++ 13.2.0, Lemonlime 0.3.4。
3. 文件名（程序名和输入输出文件名）必须使用英文小写。
4. C/C++ 中函数 `main()` 返回类型必须是 `int`，程序正常结束返回值必须是 0。
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。

数数 (count)

【题目描述】

给定正整数 N 。
求有序数对 (X,Y) 的数量，满足

- 1. $1 \leq X,Y \leq N$
- 2. X 的最高位数字等于 Y 的个位数字
- 3. Y 的最高位数字等于 X 的个位数字

【输入格式】

从文件 `count.in` 中读入数据。
一行一个正整数 N 。

【输出格式】

输出到文件 `count.out` 中。
一行一个整数，表示答案。

【样例 1 输入】

1

11

【样例 1 输出】

1

12

【样例 1 解释】

样例 1 的 12 对默契数：
 $(1,1), (1,11), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9), (11,1), (11,11)$

【样例 2】

见选手目录下的 `count/count2.in` 与 `count/count2.ans`。

【数据范围】

- 40% 数据: $N \leq 10$
- 70% 数据: $N \leq 2000$
- 100% 数据: $N \leq 2 \times 10^5$

画画 (paint)

【题目描述】

给定一幅 $n \times m$ 的原始图像，每个位置的颜色为小写字母（共 s 种颜色）。

熊孩子制作了 k 个副本，编号依次为 $1, 2, \dots, k$ 。对于第 i 个副本，他将左上角 (x_{i_1}, y_{i_1}) 到右下角 (x_{i_2}, y_{i_2}) 的矩形区域整体涂成颜色 c_i ，其余位置保持与原始图像相同。

定义两幅画 A, B 的差距为

$$\text{dis}(A, B) = \sum_{i=1}^n \sum_{j=1}^m |A_{i,j} - B_{i,j}|,$$

若选择第 t 个副本作为封面，则整本书的差距为

$$\text{Gap}(t) = \sum_{\substack{1 \leq i \leq k \\ i \neq t}} \text{dis}(\text{第}t\text{个副本}, \text{第}i\text{个副本}).$$

求最小的 $\text{Gap}(t)$ 及对应的最小封面编号 t 。

【输入格式】

从文件 `paint.in` 中读入数据。

第一行四个正整数 n, m, k, s 。

随后 n 行，每行一个长度为 m 的字符串，描述原始图像。

随后 k 行，每行五个元素： $x_1 \ y_1 \ x_2 \ y_2 \ c$ ，表示在第 i 个副本上把对应矩形涂成字符 c 。

【输出格式】

输出到文件 `paint.out` 中。

一行两个整数，依次为最小差距与对应的最小封面编号，空格分隔。

【样例 1 输入】

```
1 5 7 10 4
2 c b d a d b d
3 b b c c a d a
4 d a c d a b d
5 c d c a c b c
6 b a a d d a d
```

```
7 1 3 5 6 a
8 1 6 2 6 c
9 4 1 5 4 b
10 3 4 5 4 b
11 2 2 3 7 c
12 2 4 4 5 d
13 2 5 3 5 a
14 4 7 5 7 c
15 3 1 3 5 c
16 4 4 5 6 c
```

【样例 1 输出】

```
1 85 7
```

【样例 2】

见选手目录下的 *paint/paint2.in* 与 *paint/paint2.ans*。

【数据范围】

- $n, m \leq 1000, k \leq 300\,000, s \leq 26$
- 对于 25 % 的数据, 有 $n, m \leq 100, k \leq 350$
- 对于另外 15 % 的数据, 有 $n, m \leq 100, k \leq 1200$
- 对于另外 10 % 的数据, 有 $n, m \leq 100, k \leq 7000$
- 对于另外 10 % 的数据, 有 $\forall 1 \leq i, j \leq k, x_{i_1} = x_{j_1}, x_{i_2} = x_{j_2}, y_{i_1} = y_{j_1}, y_{i_2} = y_{j_2}$
- 对于另外 20 % 的数据, 有 $s = 2$
- 其他数据无特殊限制

打字机 (typer)

【题目描述】

给定原始字符串 S 和目标字符串 T 。

接下来给出 m 个独立询问，每个询问给出区间 $[l_i, r_i]$ 。每次询问要求：把 S 的子串 $S[l_i, r_i]$ 变成字符串 T ，只能使用古董打字机的三种操作：

- 删除一个字符；
- 在任意位置插入一个字符；
- 把一个字符替换成另一个字符。

对每个询问，输出所需的最少操作次数。

注意：所有询问彼此独立， S 不会被真正修改。

【输入格式】

从文件 **typer.in** 中读入数据。

第一行一个字符串 S 。

第二行一个字符串 T 。

第三行一个正整数 m 。

接下来 m 行，每行两个正整数 l_i, r_i ，表示询问的区间。

【输出格式】

输出到文件 **typer.out** 中。

共 m 行，每行一个正整数，表示对应询问的最少操作次数。

【样例 1 输入】

```
1 ababa
2 abc
3 4
4 1 3
5 2 3
6 2 5
7 1 5
```

【样例 1 输出】

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

【样例 1 解释】

- 将 $S[1, 3] = \text{aba}$ 变成 abc : 把第 3 个字符 a 改为 c , 操作数 1。
- 将 $S[2, 3] = \text{ba}$ 变成 abc : 把 a 改为 c 并在开头插入 a , 操作数 2。
- 将 $S[2, 5] = \text{baba}$ 变成 abc : 删除第 1 个字符并把第 3 个字符 a 改为 c , 操作数 2。
- 将 $S[1, 5] = \text{ababa}$ 变成 abc : 删除后两位并把第 3 个字符 b 改为 c , 操作数 3。

【样例 2】

见选手目录下的 *typer/typer2.in* 与 *typer/typer2.ans*。

【数据范围】

- $|S|, m \leq 10^5$, $|T| \leq 20$, 字符串仅含小写字母。
- 子任务 1 (20 分): $|S|, m \leq 20$ 。
- 子任务 2 (20 分): $|S|, m \leq 1000$ 。
- 子任务 3 (20 分): $|T| \leq 2$ 。
- 子任务 4 (40 分): 无特殊限制。

堆 (heap)

【题目描述】

给定一棵含 n 个节点的有根树，节点编号为 $1 \dots n$ ，且保证父亲编号小于子节点编号。现要把亮度为 $1, 2, \dots, n$ 的 n 盏互不相同的灯分别挂到节点 $1, 2, \dots, n$ 上，使得每个父亲节点所挂灯的亮度严格小于其所有子节点所挂灯的亮度。一个满足上述条件的灯与节点的对应关系被称为一个合法排列 p (p_i 表示挂在节点 i 上的灯亮度)。

给定一个排列 a ，求：

- 1. 在字典序不大于 a 的所有合法排列中，字典序最大的合法排列 b ；
- 2. 把所有合法排列按字典序升序排序后， b 的排名（对 $10^9 + 7$ 取模）。

【输入格式】

从文件 `heap.in` 中读入数据。
第一行一个正整数 n 。
第二行 $n - 1$ 个正整数，第 i 个数为节点 $i + 1$ 的父亲编号。
第三行 n 个正整数，表示排列 a （保证为 $1 \dots n$ 的排列）。

【输出格式】

输出到文件 `heap.out` 中。
第一行输出 n 个正整数，表示排列 b ，空格分隔。第二行输出一个整数，表示 b 的排名模 $10^9 + 7$ 。

【样例 1 输入】

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

【样例 1 输出】

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


【样例 2】

见选手目录下的 *heap/heap2.in* 与 *heap/heap2.ans*。

【数据范围】

- 所有数据： $n \leq 5000$ ，且输入的 a 为 $1 \dots n$ 的排列。
- 子任务 1（10 分）： $n \leq 10$ 。
- 子任务 2（20 分）： $n \leq 20$ 。
- 子任务 3（20 分）：对所有 $i > 1$ ，节点 i 的父亲均为 1。
- 子任务 4（20 分）： $n \leq 300$ 。
- 子任务 5（30 分）：无特殊限制。