2024 年全国青少年信息学奥林匹克联赛

CQBZ

时间: 2024 **年** 11 **月** 15 **日** 08:00 ~ 12:00

题目名称	数字	加训	排序	改造
题目类型	传统型	传统型	传统型	传统型
目录	`number`	`train`	`sort`	`modification`
可执行文件名	`number`	`train`	`sort`	`modification`
输入文件名	`number.in`	`train.in`	`sort.in`	`modification.in`
输出文件名	`number.out`	`train.out`	`sort.out`	`modification.out`
每个测试点时限	1.0 秒	2.0 秒	3.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	`number.cpp`	`train.cpp`	`sort.cpp`	`modification.cpp`

编译选项

对于 C++ 语言	`-lm -O2 -std=c++14`

注意事项与提醒（请选手务必仔细阅读）

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立**好的，且**带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
4. 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
5. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
6. 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**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式 (例如 scanf 函数) 避免出错。

12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

13. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。

14. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。15. **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**。

数字（number）

【题目描述】

小 P 终于学会了自然数！他欣喜地向阳光幼儿园的好朋友小 L 炫耀，小 L 却不以为意，头也不抬的说“好吧，我考考你，不超过自然数 n 的自然数一共有多少个?”。小 P 对小 L 的羞辱感到愤怒，还没开口，小 L 就继续问道“不超过自然数 n 的自然数中，不含数字 6 一共有多少个?”这不也是简单题，吗?

小 P 自然是不会了，但是你要解决的当然不是这样简单的问题。现在请你回答不超过自然数 n 的自然数中，不含 $\{a_m\}$ 中数字的一共有多少个。($a_i \in [0, 9]$)

【输入格式】

从文件 ***number.in*** 中读入数据。

第一行一个整数 T，表示数据组数。

接下来的每组数据共两行。

第一行两个整数 n, m，分别表示给定的自然数，以及集合 $\{a_m\}$ 的大小。

第二行 m 个整数，第 i 个元素为 a_i。

【输出格式】

输出到文件 ***number.out*** 中。

共 T 行，每行一个整数，表示该组数据的答案。

【样例 1 输入】

```
2
12 2
3 5
10 1
0
```

【样例 1 输出】

```
11
9
```

【样例 1 解释】

对于第一组数据，不超过 12 且不含 3, 5 的自然数有 $\{0, 1, 2, 4, 6, 7, 8, 9, 10, 11, 12\}$，共 11 个。

对于第二组数据，不超过 10 且不含 0 的自然数有 $\{1,2,3,4,5,6,7,8,9\}$，共 9 个。

【样例 2 输入】

```
5
92 2
7 9
16 1
9
41 1
6
6 1
1
78 3
1 4 9
```

【样例 2 输出】

```
64
16
38
6
42
```

【样例 3】

见附加文件中的 number/ex_number3.in 与 number/ex_number3.ans，该组样例满足测试点 2 的条件。

【样例 4】

见附加文件中的 number/ex_number4.in 与 number/ex_number4.ans，该组样例满足测试点 6 的条件。

【样例 5】

见附加文件中的 number/ex_number5.in 与 number/ex_number5.ans，该组样例满足测试点 10 的条件。

【数据范围】

对于所有数据，保证 $1 \le T \le 1 \times 10^5$，$0 \le n \le 10^{18}$，$1 \le m \le 9$，$0 \le a_i \le 9$。

测试点编号	n	特殊性质
1	≤ 200	无
$2 \sim 3$	$\leq 1 \times 10^{18}$	ABC
4	$\leq 1 \times 10^{18}$	AC
5	$\leq 1 \times 10^{18}$	BC
6	$\leq 1 \times 10^{18}$	AB
7	$\leq 1 \times 10^{18}$	C
8	$\leq 1 \times 10^{18}$	A
9	$\leq 1 \times 10^{18}$	B
10	$\leq 1 \times 10^{18}$	无

特殊性质 A：$1 \leq a_i \leq 9$。

特殊性质 B：$m = 1$。

特殊性质 C：保证 n 中不含 a_i。

建议使用较快的输入输出方式。

加训（train）

【题目描述】

小 P 为了准备接下来的一次比赛，给自己安排了详细的训练计划，但是由于种种不可抗因素，它的计划全部都泡汤了，最后只能在家中狭小的区域进行训练。

为了合理的利用空间，小 P 将自己的家从左往右依次分为了 n 个区域，并在每一个区域中写上了 L,R,O 中的一种字符，表示当前这个区域的状态。具体来说，若为 L，下一步应该向左移动；若为 R，下一步应该向右移动；若为 O，训练结束。

但小 P 发现自己可能会一直移动，永远无法停止，所以进行了一些修改，当从一个区域离开时，该区域上的字符翻转（即 L 变为 R，R 变为 L）。同时，小 P 也保证了第一个区域和第 n 个区域为 O。

虽然小 P 是不会感到疲惫的，但它还是想知道从每一个区域出发，走多少步之后会结束。

注意：从每一个区域出发的答案独立。

【输入格式】

从文件 ***train.in*** 中读入数据。

第一行一个正整数 n，表示区域个数。

第二行一个长度为 n 的字符串，表示初始序列。

【输出格式】

输出到文件 ***train.out*** 中。

共一行，第 i 个数字表示从区域 i 出发时走多少步会结束训练。

【样例 1 输入】

```
5
ORLLO
```

【样例 1 输出】

```
0 3 6 5 0
```

【样例 1 解释】

从五个区域出发的路径分别为：

1

$2 \to 3 \to 2 \to 1$

$3 \to 2 \to 3 \to 4 \to 3 \to 2 \to 1$

$4 \to 3 \to 2 \to 3 \to 4 \to 5$

5

【样例 2 输入】

```
12
OLLLRRLRRRLO
```

【样例 2 输出】

```
0 1 2 3 8 19 19 14 9 6 3 0
```

【样例 3】

见附加文件中的 train/ex_train3.in 与 train/ex_train3.ans，该组样例满足测试点 10 的条件。

【样例 4】

见附加文件中的 train/ex_train4.in 与 train/ex_train4.ans，该组样例满足测试点 17 的条件。

【数据范围】

对于所有数据，保证 $1 \le n \le 5 \times 10^5$。

测试点编号	n	特殊性质
$1 \sim 3$	$\le 5 \times 10^2$	无
$4 \sim 6$	$\le 1 \times 10^5$	保证 R 的数量不超过 20
$7 \sim 9$	$\le 1 \times 10^5$	不存在连续的 L 和连续的 R
$10 \sim 13$	$\le 1 \times 10^5$	只有首尾有 O，且 R 均在 L 左侧
$14 \sim 16$	$\le 5 \times 10^3$	无
$17 \sim 18$	$\le 1 \times 10^5$	无
$19 \sim 20$	$\le 5 \times 10^5$	无

排序（sort）

【题目描述】

小 L 有一个长度为 n 的排列（其中 n 为偶数），下面小 L 会对这个序列进行若干次变换。每次变换如下：

```
cnt = 0
j = n / 2 + 1

for i from 1 to n / 2:
    while j <= n and a[j] < a[i]:
        newa[++cnt] = a[j]
        j++
    newa[++cnt] = a[i]

while j <= n:
    newa[++cnt] = a[j]
    j++
```

即在归并排序中省略了向下递归的过程。现在，小 L 想知道，在经历了 t 次变换后，a_i 的值是多少。

【输入格式】

从文件 ***sort.in*** 中读入数据。

第一行两个整数 n, Q，分别表示序列长度和询问次数。

接下来一行 n 个整数 a_i，表示初始排列。

接下来 Q 行，每行两个整数 t, i，表示询问经历了 t 次变换后 a_i 的值。

【输出格式】

输出到文件 ***sort.out*** 中。

共 Q 行，第 i 行为第 i 次询问的答案。

【样例 1 输入】

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

【样例 1 输出】

```
6
3
1
```

【样例 1 解释】

第 0 次变换后的序列 $\{2,3,6,5,4,1\}$。第 1 次变换后的序列 $\{2,3,5,4,1,6\}$。第 2 次变换后的序列 $\{2,3,4,1,5,6\}$。

【样例 2 输入】

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

【样例 2 输出】

```
8
8
1
6
3
2
```

【样例 3】

见附加文件中的 sort/ex_sort3.in 与 sort/ex_sort3.ans，该组样例满足测试点 4 的条件。

【样例 4】

见附加文件中的 sort/ex_sort4.in 与 sort/ex_sort4.ans，该组样例满足测试点 18 的条件。

【数据范围】

对于所有数据，保证 $1 \le n \le 2 \times 10^5$，$1 \le Q \le 10^6$，$0 \le t \le 10^9$，$1 \le i \le n$，$n$ 为偶数，a 为排列。

测试点编号	n	特殊性质
$1 \sim 3$	$\le 1 \times 10^3$	$t \le 1 \times 10^3$
$4 \sim 6$	$\le 1 \times 10^3$	无
$7 \sim 9$	$\le 1 \times 10^5$	$a_1 = 1, a_j = n + 2 - j (j > 1)$
$10 \sim 13$	$\le 1 \times 10^5$	不同询问中 t 相同
$14 \sim 17$	$\le 1 \times 10^5$	$Q \le 1 \times 10^5$
$18 \sim 20$	$\le 2 \times 10^5$	无

改造（modification）

【题目描述】

众所周知，机器人的基因可以表示为 01 序列。在这种意义下，小 L 也可以被视为一种广义机器人。为了让自己变得更加强大，小 L 决定对自己进行基因改造。

为了比较基因的强弱（这并不违反机器人的伦理），小 L 抽象出了「价值」这一概念，定义一个 01 基因序列的「价值」为它的**最长不下降子序列**。现在小 L 想知道，如果将自己的基因序列划分为 k 个子段并进行重排，可以获得的最大「价值」是多少。

这个问题自然就交给你啦！

【输入格式】

从文件 ***modification.in*** 中读入数据。

第一行一个正整数 n，表示小 L 的基因长度。

第二行一个长度为 n 的字符串，表示小 L 基因的 01 序列。

【输出格式】

输出到文件 ***modification.out*** 中。

共一行，第 i 个数字表示 $k = i$ 时的最大「价值」。

【样例 1 输入】

```
8
01100100
```

【样例 1 输出】

```
5 7 7 8 8 8 8 8
```

【样例 1 解释】

$k = 1$ 时划分为子段 01100100，最长不下降子序列为 $\underline{01}1\underline{00}\underline{100}$。

$k = 2$ 时划分为子段 $011, 00100$，拼接为 00100011，最长不下降子序列为 $\underline{00}\underline{100011}$。

$k = 3$ 时划分为子段 $011, 001, 00$，拼接为 00001011，最长不下降子序列为 $\underline{00001011}$。

$k = 4$ 时划分为子段 $011, 00, 1, 00$，拼接为 00000111，最长不下降子序列为 $\underline{00000111}$。

$k > 4$ 时子段拼接结果与 $k = 4$ 相同。注意上面展示的划分，拼接方案与最长不下降子序列都可能不是唯一的。

【样例 2 输入】

```
20
```

```
2 10110001010110001101
```

【样例 2 输出】

```
1 12 14 14 16 16 17 17 18 18 19 19 20 20 20 20 20 20 20 20 20
```

【样例 3】

见附加文件中的 modification/ex_modification3.in 与 modification/ex_modification3.ans，该组样例满足测试点 7 的条件。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 3 \times 10^5$。

测试点编号	n
$1 \sim 3$	≤ 8
$4 \sim 6$	≤ 20
$7 \sim 9$	≤ 200
$10 \sim 13$	$\leq 2 \times 10^3$
$14 \sim 17$	$\leq 8 \times 10^4$
$18 \sim 20$	$\leq 3 \times 10^5$