

2024 CCF 非专业级软件能力认证

CSP-S 2024 第二轮认证

模拟试题

时间：2024 年 8 月 2 日 08:00 ~ 12:00

题目名称	魔法阵	石头	信息传递	树
题目类型	传统型	传统型	传统型	传统型
目录	magic	stone	transfer	tree
可执行文件名	magic	stone	transfer	tree
输入文件名	magic.in	stone.in	transfer.in	tree.in
输出文件名	magic.out	stone.out	transfer.out	tree.out
每个测试点时限	1.0 秒	1.0 秒	3.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	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	magic.cpp	stone.cpp	transfer.cpp	tree.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。

1、魔法阵 (magic)

【题目描述】

六十年一次的魔法战争就要开始了，大魔法师准备从附近的魔法场中汲取魔法能量。

大魔法师有 m 个魔法物品，编号分别为 $1, 2, \dots, m$ 。每个物品具有一个魔法值，我们用 X_i 表示编号为 i 的物品的魔法值。每个魔法值 X_i 是不超过 n 的正整数，可能有多个物品的魔法值相同。

大魔法师认为，当且仅当四个编号为 a, b, c, d 的魔法物品满足 $X_a < X_b < X_c < X_d$, $X_b - X_a = 2(X_d - X_c)$ ，并且 $X_b - X_a < (X_c - X_b)/3$ 时，这四个魔法物品形成了一个魔法阵，他称这四个魔法物品分别为这个魔法阵的 A 物品，B 物品，C 物品，D 物品。

现在，大魔法师想要知道，对于每个魔法物品，作为某个魔法阵的 A 物品出现的次数，作为 B 物品的次数，作为 C 物品的次数，和作为 D 物品的次数。

【输入格式】

第一行包含两个空格隔开的正整数 n, m 。

接下来 m 行，每行一个正整数，第 $i+1$ 行的正整数表示 X_i ，即编号为 i 的物品的魔法值。

保证 $1 \leq n \leq 15000$, $1 \leq m \leq 40000$, $1 \leq X_i \leq n$ 。每个 X_i 是分别在合法范围内等概率随机生成的。

【输出格式】

共 m 行，每行 4 个整数。第 i 行的 4 个整数依次表示编号为 i 的物品作为 A, B, C, D 物品分别出现的次数。

保证标准输出中的每个数都不会超过 10^9 。每行相邻的两个数之间用恰好一个空格隔开。

【样例 1 输入】

```
30 8
1
24
7
28
5
29
26
24
```

【样例 1 输出】

```
4 0 0 0
0 0 1 0
0 2 0 0
0 0 1 1
1 3 0 0
```

```

0 0 0 2
0 0 2 2
0 0 1 0

```

【样例 1 解释】

共有 5 个魔法阵，分别为：

- 物品 1, 3, 7, 6，其魔法值分别为 1, 7, 26, 29；
- 物品 1, 5, 2, 7，其魔法值分别为 1, 5, 24, 26；
- 物品 1, 5, 7, 4，其魔法值分别为 1, 5, 26, 28；
- 物品 1, 5, 8, 7，其魔法值分别为 1, 5, 24, 26；
- 物品 5, 3, 4, 6，其魔法值分别为 5, 7, 28, 29。

以物品 5 为例，它作为 A 物品出现了 1 次，作为 B 物品出现了 3 次，没有作为 C 物品或者 D 物品出现，所以这一行输出的四个数依次为 1, 3, 0, 0。

此外，如果我们将输出看作一个 m 行 4 列的矩阵，那么每一列上的 m 个数之和都应等于魔法阵的总数。所以，如果你的输出不满足这个性质，那么这个输出一定不正确。你可以通过这个性质在一定程度上检查你的输出的正确性。

【样例 2】

见选手目录下的 magic/magic2.in 与 magic/magic2.ans。

【数据范围】

编号	$n=$	$m=$
1	10	12
2	15	18
3	20	25
4	30	35
5	40	50
6	50	70
7	65	100
8	80	125
9	100	150
10	125	200
11	150	250
12	200	350
13	250	500
14	350	700
15	500	1000
16	700	2000
17	1000	5000
18	2000	10000
19	5000	20000
20	15000	40000

2、石头 (stone)

【题目描述】

桌子上有 n 个石头围成一个环。每个石头都有一种颜色。每种颜色可以用不同的小写英文字母表示，所以总共有 26 种颜色。不同的石头可能有相同的颜色。如果每一对相邻的石头都是不同颜色的，则称这 n 个石头构成的环是美丽的。

两个石头是相邻的充要条件是这两个石头中间没有其他石头。例如：1 号和 2 号是相邻的，2 号和 3 号是相邻的， \dots ， n 号和 1 号是相邻的。现在，你可以从这 n 个石头中拿走一段连续的石头（可以为空），且你只能拿一次。

你的任务是对于每个 k ($0 \leq k \leq n-1$)，判断是否存在一种取石头的方案，使得在拿走 k 个连续的石头后，剩下的 $n-k$ 个石头构成的环是美丽的。

【输入格式】

输入包含多组测试数据，以 EOF 结束。

每组数据有一行字符串 s ，字符串第 i 位表示桌上第 i 个石头的颜色。设字符串 s 的长度为 n ，则 $1 \leq n \leq 10^6$ 。字符串只包含小写英文字母。

【输出格式】

对于每组数据，按照样例的格式输出数据编号和一个长度为 n 的字符串。如果存在一种取石头的方案，使得在拿走 k 个连续的石头后，剩下的 $n-k$ 个石头构成的环是美丽的，则字符串的第 k 位（由 0 开始编号）为 1，否则为 0。

【样例 1 输入】

```
rrg
rrrrr
brbg
abab
```

【样例 1 输出】

```
Case 1:011
Case 2:00001
Case 3:1111
Case 4:1011
```

【数据范围】

对于所有数据，数据组数不超过 6 组， $1 \leq n \leq 10^6$ 。

对于 20%的数据： $n \leq 20$

对于另外 20%的数据：字符串随机生成

对于另外 20%的数据： $n \leq 1000$

对于剩下 40%的数据： $n \leq 500000$

3、信号传递 (transfer)

【题目描述】

一条道路上从左至右排列着 m 个信号站，初始时从左至右依次编号为 $1, 2, \dots, m$ ，相邻信号站之间相隔 1 单位长度。每个信号站只能往它右侧的任意信号站传输信号（称为普通传递），每单位长度距离需要消耗 1 单位时间。道路的最左侧有一个控制塔，它的最左侧信号站的左侧，与其相隔 1 单位长度。控制塔能与任意信号站进行双向信号传递（称为特殊传递），但每单位长度距离需要消耗 k 个单位时间。对于给定的长度为 n 的信号传递序列 S ，传递规则如下：

- 1. 共 $n-1$ 次信号传递，第 i 次信号传递将把信号从 S_i 号信号站传递给 S_{i+1} 号。
- 2. 若 S_{i+1} 号信号站在 S_i 号右侧，则将使用普通传递方式，从 S_i 号直接传递给 S_{i+1} 号。
- 3. 若 S_{i+1} 号信号站在 S_i 号左侧，则将使用特殊传递方式，信号将从 S_i 号传递给控制塔，再由控制塔传递给 S_{i+1} 号。
- 4. 若 $S_i=S_{i+1}$ ，则信号无须传递。

阿基作为大工程师，他能够任意多次交换任意两个信号站的位置，即他能够重排信号站的顺序，这样会使得 S 消耗的传递时间改变。现在阿基想知道，在他重排信号站顺序后 S 所消耗的传递时间最小能是多少。

【输入格式】

第一行三个整数 n, m, k ，分别代表信号传递序列 S 的长度，信号站个数，特殊传递单位长度距离的时间消耗。

第二行 n 个整数，第 i 个整数表示 S_i 。

【输出格式】

仅一行一个整数表示答案。

【输入输出样例 1】

transfer.in	transfer.out
3 3 1	2
1 2 3	

【样例 1 说明】

信号站顺序保持不变，两次使用普通传递方式，时间消耗为 $1+1=2$ 。

【输入输出样例 2】

transfer.in	transfer.out
4 3 1	6
1 2 3 1	

【样例 2 说明】

对于排列 1, 2, 3, 传递时间为 $1+1+(3\times 1+1\times 1)=6$ 。

对于排列 1, 3, 2, 传递时间为 $2+(3\times 1+2\times 1)+(2\times 1+1\times 1)=10$ 。

对于排列 2, 1, 3, 传递时间为 $(2\times 1+1\times 1)+2+(3\times 1+2\times 1)=10$ 。

对于排列 2, 3, 1, 传递时间为 $(3\times 1+1\times 1)+1+1=6$ 。

对于排列 3, 1, 2, 传递时间为 $1+(3\times 1+1\times 1)+1=6$ 。

对于排列 3, 2, 1, 传递时间为 $(3\times 1+2\times 1)+(2\times 1+1\times 1)+2=10$ 。

【样例 3】

见选手目录下的 transfer/transfer3.in 与 transfer/transfer3.ans。

【数据范围】

对于 30%的数据： $m\leq 8$ ； $n\leq 100$ 。

对于 60%的数据： $m\leq 20$ 。

对于 70%的数据： $m\leq 21$ 。

对于 80%的数据： $m\leq 22$ 。

对于 100%的数据： $2\leq m\leq 23$ ； $2\leq n\leq 10^5$ ； $1\leq k\leq 100$ ； $1\leq S_i\leq m$ 。

4、树 (tree)

【题目描述】

给定一棵 n 个结点的有根树 T ，结点从 1 开始编号，根结点为 1 号结点，每个结点有一个正整数权值 v_i 。

设 x 号结点的子树内（包含 x 自身）的所有结点编号为 c_1, c_2, \dots, c_k ，定义 x 的价值为：

$$val(x) = (v_{c_1} + d(c_1, x)) \oplus (v_{c_2} + d(c_2, x)) \oplus \dots \oplus (v_{c_k} + d(c_k, x))$$

其中 $d(x, y)$ 表示树上 x 号结点与 y 号结点间唯一简单路径所包含的边数， $d(x, x) = 0$ 。 \oplus 表示异或运算。

请你求出以下表达式的结果：

$$\sum_{i=1}^n val(i)$$

【输入格式】

第一行一个正整数 n 表示树的大小。

第二行 n 个正整数表示 v_i 。

接下来一行 $n-1$ 个正整数依次表示 2 号结点到 n 号结点，每个结点的父亲编号 p_i 。

【输出格式】

仅一行一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
12
```

【样例 1 解释】

$val(1) = (5+0) \oplus (4+1) \oplus (1+1) \oplus (2+2) \oplus (3+2) = 3。$

$val(2) = (4+0) \oplus (2+1) \oplus (3+1) = 3。$

$val(3) = (1+0) = 1。$

$val(4) = (2+0) = 2。$

$val(5) = (3+0) = 3。$

【数据范围】

对于 10% 的数据： $1 \leq n \leq 2501$

对于 40% 的数据： $1 \leq n \leq 152501$

另有 20% 的数据：所有 $p_i = i-1$ ($2 \leq i \leq n$)

另有 20% 的数据：所有 $v_i = 1$ ($1 \leq i \leq n$)

对于 100% 的数据： $1 \leq n, v_i \leq 525010, 1 \leq p_i \leq n。$