

# 洛谷 CSP-S 2024 模拟赛

## KDOI Round 10

### 提高级

时间：2024 年 10 月 13 日 14:30 ~ 18:30

题目名称	商店砍价	水杯降温	反回文串	超级演出
题目类型	传统型	传统型	传统型	传统型
目录	<b>bargain</b>	<b>water</b>	<b>anti</b>	<b>show</b>
可执行文件名	<b>bargain</b>	<b>water</b>	<b>anti</b>	<b>show</b>
输入	标准输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	25	25	20	25
测试点是否等分	是	是	是	是
样例数目	5	7	2	6

提交源程序文件名

对于 C++ 语言	<b>bargain.cpp</b>	<b>water.cpp</b>	<b>anti.cpp</b>	<b>show.cpp</b>
-----------	--------------------	------------------	-----------------	-----------------

编译选项

对于 C++ 语言	<b>-O2 -std=c++14</b>
-----------	-----------------------

注意事项（请仔细阅读）

1. 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。
2. 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）。
3. 选手提交的程序源文件必须不大于 100 KB。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 只提供 Linux 格式附加样例文件。
6. 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）和其他可能造成不公平的方法。
7. 所有题目均使用**标准输入输出**。

## 商店砍价 (bargain)

### 【题目描述】

有一个正整数  $n$ ，保证其只由数字  $1 \sim 9$  构成。

你可以做任意多次如下操作：

- 选择  $n$  的一个数位  $x$ ，花费  $v_x$  的代价删除它，注意，此时  $n$  的数位个数会减少 1， $n$  的值也会发生相应的变化；
  - 或者，花费  $n$  的代价把剩余的所有数位删除。
- 求把整个数删除的最小代价。

### 【输入格式】

从标准输入读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数  $c$ ，表示测试点编号。 $c = 0$  表示该测试点为样例。

第二行包含一个正整数  $t$ ，表示测试数据组数。

对于每组测试数据：

- 第一行一个正整数  $n$ ，表示这个数的初始值。
- 第二行九个正整数  $v_1, v_2, \dots, v_9$ ，表示删除每个数位的代价。

### 【输出格式】

输出到标准输出。

对于每组测试数据：

- 输出一行一个正整数，表示最小代价。

### 【样例 1 输入】

```
1 0
2 3
3 123
4 10 10 10 10 10 10 10 10 10
5 1121
6 2 1 2 2 2 2 2 2 2
7 987654321
8 1 2 3 4 5 6 7 8 9
```

### 【样例 1 输出】

```
1 21
2 6
3 45
```

### 【样例 1 解释】

对于第一组测试数据，最优操作方案如下：

- 删除数位 2，代价为 10，此时  $n$  变为 13；
- 删除数位 3，代价为 10，此时  $n$  变为 1；
- 删除  $n$  的剩余所有数位，代价为 1。

总代价为  $10 + 10 + 1 = 21$ ，可以证明，这是代价的最小值。

对于第二组测试数据，一种最优操作方案如下：

- 删除第一个数位 1，代价为 2，此时  $n$  变为 121；
- 删除最后一个数位 1，代价为 2，此时  $n$  变为 12；
- 删除数位 2，代价为 1，此时  $n$  变为 1；
- 删除  $n$  的剩余所有数位，代价为 1。

总代价为  $2 + 2 + 1 + 1 = 6$ 。

### 【样例 2】

见选手目录下的 *bargain/bargain2.in* 与 *bargain/bargain2.ans*。

这个样例满足测试点 3 ~ 6 的约束条件。

### 【样例 3】

见选手目录下的 *bargain/bargain3.in* 与 *bargain/bargain3.ans*。

这个样例满足测试点 11 的约束条件。

### 【样例 4】

见选手目录下的 *bargain/bargain4.in* 与 *bargain/bargain4.ans*。

这个样例满足测试点 17, 18 的约束条件。

### 【样例 5】

见选手目录下的 *bargain/bargain5.in* 与 *bargain/bargain5.ans*。

这个样例满足测试点 23 ~ 25 的约束条件。

【数据范围】

对于全部的测试数据，保证：

- $1 \leq t \leq 10$ ;
- $1 \leq n < 10^{10^5}$ ;
- 对于任意  $1 \leq i \leq 9$ ,  $1 \leq v_i \leq 10^5$ ;
- $n$  由数字  $1 \sim 9$  构成。

测试点	$n <$	$v_i \leq$	特殊性质
1	100	$10^5$	无
2	$10^3$		
3 ~ 6	$10^{18}$		
7 ~ 9	$10^{40}$		
10	$10^{10^5}$		$n$ 由至多一种数字构成
11			$n$ 由至多两种数字构成
12, 13			$n$ 由至多三种数字构成
14 ~ 16	$10^{10^3}$		$v_1 = v_2 = v_3 = \cdots = v_9$
17, 18	$10^{10^5}$		
19, 20	$10^{100}$	100	无
21, 22	$10^{10^3}$	$10^3$	
23 ~ 25	$10^{10^5}$	$10^5$	

## 水杯降温 (water)

### 【题目描述】

小 S 有一棵包含  $n$  个节点的有根树，且根为节点 1。节点  $i$  ( $1 \leq i \leq n$ ) 上放置了一个初始水温为  $a_i$  的水杯。

在不知道水温的情况下拿起水杯喝水并被烫了  $\text{inf}$  次的小 S 决定将这些水杯的水温全部变为 0 后再喝它们。

现在，小 S 可以分别进行以下两种操作任意次：

- 使用一个在节点  $i$  的加热装置。这会使以  $i$  为根的子树内所有水杯里的水温均增加 1；
- 或者，从某个叶子节点  $i$  向根方向吹一阵风。这会使  $i$  到根所有水杯里的水温均减少 1。

请你帮小 S 判断：能否将所有节点上的水杯的水温都变为 0。

### 【输入格式】

从标准输入读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数  $c$ ，表示测试点编号。 $c = 0$  表示该测试点为样例。

第二行包含一个正整数  $t$ ，表示测试数据组数。

对于每组测试数据：

- 第一行包含一个正整数  $n$ ，表示节点数量。
- 第二行  $n - 1$  个正整数  $f_2, \dots, f_n$ ，其中  $f_i$  表示节点  $i$  的父亲节点编号。保证  $f_i < i$ 。
- 第三行  $n$  个正整数  $a_i$ ，表示初始水温。

### 【输出格式】

输出到标准输出。

对于每组测试数据：

- 如果可以将水温变为 0，输出一行一个字符串 **Huoyu**；
- 如果无法将水温变为 0，输出一行一个字符串 **Shuiniao**。

### 【样例 1 输入】

```
1 0
2 5
3 5
```

```
4 1 1 2 3
5 6 5 2 2 1
6 5
7 1 1 2 2
8 6 5 1 2 1
9 5
10 1 1 2 2
11 4 -1 5 -2 -2
12 5
13 1 1 2 2
14 6 -4 8 -3 -3
15 5
16 1 1 2 2
17 -1 -1 -1 -4 -1
```

### 【样例 1 输出】

```
1 Shuiniiao
2 Huoyu
3 Shuiniiao
4 Shuiniiao
5 Huoyu
```

### 【样例 1 解释】

记  $A_u$  表示在节点  $u$  使用加热装置的操作,  $B_u$  表示从节点  $u$  吹一阵风的操作,  $(S)^k$  表示将操作序列  $S$  重复  $k$  次。

- 对于第一、三、四组测试数据, 可以证明, 小 S 无法将所有水杯的水温都变为 0;
- 对于第二组测试数据, 一种可能的操作序列为:  $B_3(A_4)^3(B_4)^5B_5$ ;
- 对于第五组测试数据, 一种可能的操作序列为:  $(A_4)^3A_1$ 。

### 【样例 2】

见选手目录下的 *water/water2.in* 与 *water/water2.ans*。

这个样例满足测试点 3 的约束条件。

**【样例 3】**

见选手目录下的 *water/water3.in* 与 *water/water3.ans*。  
这个样例满足测试点 7, 8 的约束条件。

**【样例 4】**

见选手目录下的 *water/water4.in* 与 *water/water4.ans*。  
这个样例满足测试点 12 的约束条件。

**【样例 5】**

见选手目录下的 *water/water5.in* 与 *water/water5.ans*。  
这个样例满足测试点 13, 14 的约束条件。

**【样例 6】**

见选手目录下的 *water/water6.in* 与 *water/water6.ans*。  
这个样例满足测试点 15 ~ 17 的约束条件。

**【样例 7】**

见选手目录下的 *water/water7.in* 与 *water/water7.ans*。  
这个样例满足测试点 18 ~ 21 的约束条件。

**【数据范围】**

记  $\sum n$  为单个测试点内所有测试数据中  $n$  的和。

对于全部的测试数据，保证：

- $1 \leq t \leq 1\,000$ ;
- $2 \leq n \leq 10^5$ ,  $\sum n \leq 10^6$ ;
- 对于任意  $2 \leq i \leq n$ ,  $1 \leq f_i < i$ ;
- 对于任意  $1 \leq i \leq n$ ,  $-10^{12} \leq a_i \leq 10^{12}$ 。

测试点	$n \leq$	$\sum n \leq$	$ a_i  \leq$	特殊性质
1	5	50	5	无
2		200		
3		5 000		
4, 5	50	500	50	
6			$10^8$	
7, 8	200	2 000	200	
9			$10^8$	
10, 11	1 000	$10^4$	1 000	
12			$10^8$	
13, 14	$10^5$	$3 \times 10^5$	$10^{12}$	A
15 ~ 17				B
18 ~ 21				C
22, 23	$3 \times 10^4$	$10^5$	$10^8$	无
24, 25	$10^5$	$10^6$	$10^{12}$	

- 特殊性质 A：对于任意  $1 \leq i \leq n$ ， $f_i = i - 1$ ；
- 特殊性质 B：对于任意  $1 \leq i \leq n$ ， $a_i \leq \left(\sum_{f_j=i} a_j\right) + 5$ ，其中设  $f_1 = 0$ ；
- 特殊性质 C：树的深度不超过 2，其中深度指所有节点到根的边数中的最大值。

【提示】

本题输入输出量较大，请使用适当的 I/O 方式。



## 反回文串 (anti)

### 【题目描述】

我们称一个长度为  $m$  的字符串  $r$  是回文的, 当且仅当  $r_i = r_{m+1-i}$  对所有  $1 \leq i \leq m$  均成立。

给定一个长度为  $n$  的字符串  $s$ , 你需要把  $s$  分成若干个非空子序列, 使得每一个子序列都**不是**回文的, 并最大化划分成的子序列数。

形式化地说, 你需要给出一组序列  $(a_1, a_2, \dots, a_k)$ , 满足:

- 对于任意  $1 \leq i \leq k$ , 记  $l_i$  为  $a_i$  的长度, 则  $l_i \geq 1$ , 且  $1 \leq a_{i,1} < a_{i,2} < \dots < a_{i,l_i} \leq n$ ;
- 对于任意  $1 \leq i \leq n$ , 恰好存在一个二元组  $(p, q)$ , 使得  $a_{p,q} = i$ ;
- 对于任意  $1 \leq i \leq k$ , 记字符串  $t = s_{a_{i,1}} s_{a_{i,2}} \dots s_{a_{i,l_i}}$ , 则  $t$  不是回文的。

在此基础上, 你需要最大化  $k$  的值; 或者判断不存在一种合法的方案。

特别地, 如果  $k$  的值不是最大的, 你也可能获得一定的部分分。

### 【输入格式】

从标准输入读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数  $c$ , 表示测试点编号。  $c = 0$  表示该测试点为样例。

第二行包含一个正整数  $q$ , 表示测试数据组数。

对于每组测试数据:

- 第一行包含一个正整数  $n$ , 表示字符串  $s$  的长度;
- 第二行包含一个长度为  $n$  的字符串  $s$ 。保证  $s$  中仅包含小写英文字母。

### 【输出格式】

输出到标准输出。

对于每组测试数据:

- 如果不存在一种合法的方案, 输出一行一个字符串 **Shuiniiao**;
- 否则, 你需要:
  - 在第一行输出一个字符串 **Huoyu**;
  - 第二行输出一个正整数  $k$  ( $1 \leq k \leq n$ ), 表示你划分成的子序列个数;
  - 接下来  $k$  行, 对于第  $i$  行 ( $1 \leq i \leq k$ ):
    - \* 首先输出一个正整数  $l_i$  ( $1 \leq l_i \leq n$ ), 表示第  $i$  个子序列的长度;
    - \* 接下来输出  $l_i$  个正整数  $a_{i,1}, a_{i,2}, \dots, a_{i,l_i}$  ( $1 \leq a_{i,j} \leq n$ ), 表示第  $i$  个子序列。

请注意, 你的输出需要满足题目描述中所有的限制, 否则, 你将会得到 0 分。

## 【样例 1 输入】

```
1 0
2 4
3 4
4 kdoi
5 7
6 ccccccc
7 7
8 sszcdjr
9 7
10 abacaca
```

## 【样例 1 输出】

```
1 Huoyu
2 2
3 2 1 2
4 2 3 4
5 Shuiniao
6 Huoyu
7 3
8 3 1 2 3
9 2 4 5
10 2 6 7
11 Huoyu
12 3
13 2 1 4
14 3 2 3 5
15 2 6 7
```

## 【样例 1 解释】

对于第一组测试数据，显然输出构成一个合法的子序列划分，并且

- 对于第一个子序列， $t = kd$  不是回文的；
- 对于第二个子序列， $t = oi$  不是回文的。

故这是一组合法的输出。可以证明，对于这组测试数据，2 是  $k$  的最大可能值。

对于第二组数据，它的任意一个子序列都是回文的，故显然不存在合法的划分方案。

【样例 2】

见选手目录下的 *anti/anti2.in* 与 *anti/anti2.ans*。

这个样例共有 10 组数据，均满足  $n = 1000$ 。其中第 1 ~ 3 组数据满足特殊性质 A，第 4 ~ 6 组数据满足特殊性质 B。

【评分方式】

本题共有 20 个测试点，每个测试点满分 5 分。

本题采用自定义校验器 (special judge) 评测。每组测试数据可能有多组解，你只需要给出任意一组。

在每个测试点中，你的得分是在所有测试数据上得分的最小值。对于每组测试数据：

- 如果你错误地判断了是否有解或者给出了一组不合法的序列，你将会获得 0 分；
- 如果你正确判断了是否有解，并在有解时给出了一组合法的序列：
  - 如果  $k$  的值不是最大的，你将会获得 2 分；
  - 如果  $k$  的值是最大的，你将会获得 5 分。

【数据范围】

对于全部的测试数据，保证：

- $1 \leq q \leq 10$ ；
- $1 \leq n \leq 10^5$ ；
- $s$  中仅包含小写英文字母。

测试点	$n \leq$	特殊性质
1, 2	5	无
3 ~ 5	18	
6 ~ 8	1 000	B
9 ~ 11		无
12 ~ 14	$10^5$	A
15 ~ 17		B
18 ~ 20		无

- 特殊性质 A：保证  $n$  是偶数，且  $s$  中每个字符的出现次数都不超过  $\frac{n}{2}$ ；
- 特殊性质 B：保证  $s$  中仅有 **a** 和 **b**。

## 【如何使用校验器】

为了方便选手测试，在选手文件的 `anti` 目录下我们下发了 `checker.cpp` 文件作为样例校验器，选手可以编译该程序，并使用它校验自己的输出文件的结果是否合法。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：

```
1 g++ -o checker -std=c++14 -O2 checker.cpp
```

`checker` 的使用方式为：

```
1 checker <input-file> <output-file>
```

其中，参数 `<input-file>` 与 `<output-file>` 依次表示输入文件与你的输出文件。

若你的输出中的数字大小范围不合法，则校验器会给出相应提示并立即退出。否则，校验器输出以下内容：

- 在第  $i$  行 ( $1 \leq i \leq q$ ) 中，输出第  $i$  组测试数据的详细提示信息；
- 在第  $(q+1)$  行，输出这个测试点的总结信息。

例如，对于样例 1 的输入与输出，校验器将会向屏幕打印如下内容：

```
1 Test case 1: OK. Participant's answer is YES (Huoyu), and k=2.
2 Test case 2: OK. Participant's answer is NO (Shuiniiao).
3 Test case 3: OK. Participant's answer is YES (Huoyu), and k=3.
4 Test case 4: OK. Participant's answer is YES (Huoyu), and k=3.
5 ok 4 / 4 test cases passed. (4 test cases)
```

若将输出改为如下：

```
1 Huoyu
2 2
3 2 1 2
4 2 3 4
5 Huoyu
6 1
7 7 1 2 3 4 5 6 7
8 Huoyu
9 3
10 3 1 2 3
11 2 4 5
12 2 6 7
13 Huoyu
```

```
14 3
15 2 1 4
16 3 2 3 5
17 2 6 7
```

则会向屏幕打印如下内容:

```
1 Test case 1: OK. Participant's answer is YES (Huoyu), and k=2.
2 Test case 2: Wrong answer. The string t obtained in the
  subsequence a[1] is palindrome.
3 Test case 3: OK. Participant's answer is YES (Huoyu), and k=3.
4 Test case 4: OK. Participant's answer is YES (Huoyu), and k=3.
5 wrong answer 3 / 4 test cases passed.
```

**请注意:** 样例校验器只会检查你的输出是否合法, 而不会:

- 检查有解性是否判断正确;
- 检查  $k$  是否被最大化。

例如, 将样例 1 的输出改为如下:

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

此时, 样例校验器仍会返回 **ok** 的检查结果。

## 超级演出 (show)

### 【题目描述】

巡准备了一场超级演出。舞台和候场室可以看作一个包含  $n$  个点  $m$  条边的有向图，并且这个图当中没有环，也就是说，这是一张有向无环图 (DAG)。

舞台为 1 号节点，**保证所有节点均有到达节点 1 的路径**。其余的节点均为候场室，每个候场室恰有一个剧团进行等待。

巡可以对一个候场室  $u$  发布出场命令：

- 如果这个候场室的剧团还没有出场，并且存在一条  $u \rightarrow 1$  的路径上没有其余候场的剧团。那么这个剧团就会沿着这条路径到达舞台进行演出，随后退场。注意：**一个剧团退场后不会重新回到候场室**。
- 否则，这个命令被认为是无效的。

巡有一个命令序列  $a_1, a_2, \dots, a_k$  和  $q$  次询问，每次给出一个区间  $[l, r]$ 。巡想要知道如果依次对候场室  $a_l, a_{l+1}, \dots, a_r$  发布出场命令后，候场室还会剩下多少剧团等待演出。

注意：每次询问相互独立，也就是说，每次询问之前，每个候场室都恰有一个剧团进行等待。

### 【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数  $c$ ，表示测试点编号。 $c = 0$  表示该测试点为样例。

第二行包含四个正整数  $n, m, k, q$ ，表示图的点数，边数，序列长度，和询问次数。

接下来  $m$  行，每行两个正整数  $u, v$ ，表示一条从  $u$  到  $v$  的有向边。保证无自环，无重边。

接下来的一行， $k$  个正整数  $a_1, a_2, \dots, a_k$ 。

接下来  $q$  行，每行两个正整数  $l, r$ ，表示一次询问的区间。

### 【输出格式】

输出到标准输出。

$q$  行，每行一个非负整数，表示一次询问的答案。

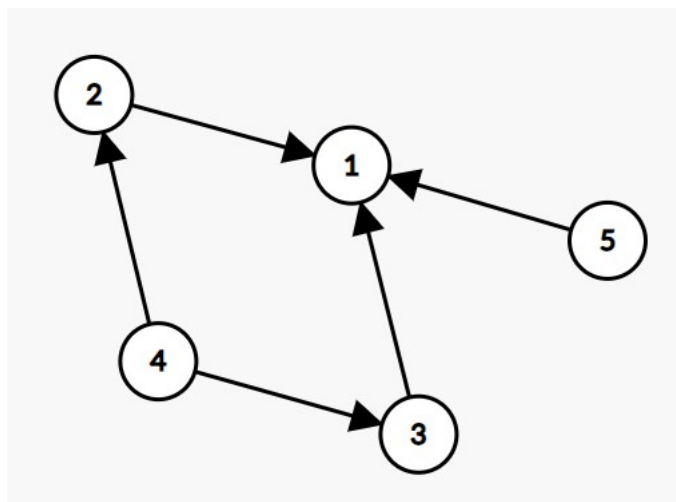
### 【样例 1 输入】

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

```
5 5 1
6 4 2
7 4 3
8 2 4 4 3 5
9 1 2
10 1 5
11 3 5
12 2 3
```

**【样例 1 输出】**

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

**【样例 1 解释】**

如图：

- 当询问  $l = 1, r = 2$  时：
  - 发布出场命令  $a_1 = 2$ 。2 沿着  $2 \rightarrow 1$  出场。
  - 发布出场命令  $a_2 = 4$ 。4 沿着  $4 \rightarrow 2 \rightarrow 1$  出场。

此时余下 3, 5 两个剧团，输出 2。

- 当询问  $l = 2, r = 3$  时：

- 发布出场命令  $a_2 = 4$ 。找不到  $4 \rightarrow 1$  的且路上没有别的剧团的路线，该指令被认为无效。
- 发布出场命令  $a_3 = 4$ 。找不到  $4 \rightarrow 1$  的且路上没有别的剧团的路线，该指令被认为无效。

此时余下 2, 3, 4, 5 四个剧团，输出 4。

### 【样例 2】

见选手目录下的 *show/show2.in* 与 *show/show2.ans*。

这个样例满足测试点 1, 2 的限制条件。

### 【样例 3】

见选手目录下的 *show/show3.in* 与 *show/show3.ans*。

这个样例满足测试点 5 ~ 8 的限制条件。

### 【样例 4】

见选手目录下的 *show/show4.in* 与 *show/show4.ans*。

这个样例满足测试点 9 ~ 11 的限制条件。

### 【样例 5】

见选手目录下的 *show/show5.in* 与 *show/show5.ans*。

这个样例满足测试点 12, 13 的限制条件。

### 【样例 6】

见选手目录下的 *show/show6.in* 与 *show/show6.ans*。

这个样例满足测试点 18, 19 的限制条件。

### 【数据范围】

对于全部的测试数据，保证：

- $1 \leq n, k, q \leq 2 \times 10^5$ ;
- $1 \leq m \leq 4 \times 10^5$ ;
- $1 \leq v < u \leq n$ ，且不存在两组相同的  $(u, v)$ ;
- 对于任意  $1 \leq i \leq k$ ， $2 \leq a_i \leq n$ ;
- 对于每组询问， $1 \leq l \leq r \leq k$ ;
- 输入构成一张有向无环图，且所有节点均存在到达节点 1 的路径。



测试点	$n, k, q \leq$	$m \leq$	特殊性质
1, 2	300	600	无
3, 4	2 000	4 000	A
5 ~ 8			无
9 ~ 11	$2 \times 10^5$	$4 \times 10^5$	A
12, 13			BC
14, 15			C
16, 17			BD
18, 19			D
20 ~ 22			B
23 ~ 25			无

- 特殊性质 A: 图退化为一棵内向树, 也就是说, 除节点 1 外, 每个点恰有一条出边, 节点 1 没有出边;
- 特殊性质 B: 保证对于每组询问,  $r = k$ ;
- 特殊性质 C: 保证对于任意  $1 \leq i < j \leq k$ ,  $a_i \neq a_j$ ;
- 特殊性质 D: 保证每个点的入度和出度均不超过 30。