CSP-S 2024 Luogu Mock Contest

KDOI Round 10

Senior

Time: 2024/10/13 14:30 ~ 18:30

Name	Bargain	Water	Anti-palind	Super Show
Type	Standard	Standard	Standard	Standard
Input	Standard Input	Standard Input	Standard Input	Standard Input
Output	Standard Output	Standard Output	Standard Output	Standard Output
Time Limit	1.0 sec	2.0 sec	1.0 sec	1.0 sec
Memory Limit	512 MiB	512 MiB	512 MiB	512 MiB
# of tests	25	25	20	25
Equally Scored	Yes	Yes	Yes	Yes
# of samples	5	7	2	6

Compilation Options

For C++	`-O2 -std=c++14`

Notice

1. The return value of `main()` must be 0.
2. The size of the source code must not exceed 100 KB.
3. The stack limit is the same as the memory limit.
4. Only Linux format sample files are provided.
5. It is strictly forbidden to use `#pragma` or `__asm__` in the source code.
6. All the problems use standard input and output.

Bargain（bargain）

【Description】

There is an integer n consisting only of digits $1 \sim 9$.

You can do an arbitrary number of operations on n (possibly zero):

- Choose a digit of n and delete it. Suppose the digit is x, this operation will cost v_x. Note that after this operation, the length of n decreases by 1, and the value of n also changes;
- Delete all the remaining digits of n. This operation will cost n.

Find the minimum cost to delete all digits of n.

【Input Format】

Read from the standard input.

Each test contains multiple test cases.

The first line of the input contains a single integer c —the id of the test. $c = 0$ represents that this is a sample test.

The second line contains a single integer t —the number of test cases.

For each test case:

- The first line contains a single integer n —the initial value of the integer.
- The second line contains nine integers $v_1, v_2, \ldots, v_9$ —the cost of deleting each digit.

【Output Format】

Write to the standard output.

For each test case:

- Print a single integer in the only line of the output —the minimum cost.

【Sample Input 1】

```
0
3
123
10 10 10 10 10 10 10 10 10
1121
2 1 2 2 2 2 2 2 2
987654321
```

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

【Sample Output 1】

```
21
6
45
```

【Sample Explanation 1】

In the first test case, the optimal operations are:

- Delete digit 2 with a cost of 10. After that, n becomes 13;
- Delete digit 3 with a cost of 10. After that, n becomes 1;
- Delete all the remaining digits of n with a cost of 1.

The total cost is $10 + 10 + 1 = 21$. It can be shown that this is the minimum cost.

In the second test case, the optimal operations are:

- Delete the first digit 1 with a cost of 2. After that, n becomes 121;
- Delete the last digit 1 with a cost of 2. After that, n becomes 12;
- Delete digit 2 with a cost of 1. After that, n becomes 1;
- Delete all the remaining digits of n with a cost of 1.

The total cost is $2 + 2 + 1 + 1 = 6$.

【Sample 2】

See ***bargain/bargain2.in*** and ***bargain/bargain2.ans*** in the user's home path.

This sample satisfies the constraints of test $3 \sim 6$.

【Sample 3】

See ***bargain/bargain3.in*** and ***bargain/bargain3.ans*** in the user's home path.

This sample satisfies the constraints of test 11.

【Sample 4】

See ***bargain/bargain4.in*** and ***bargain/bargain4.ans*** in the user's home path.

This sample satisfies the constraints of test $17, 18$.

【Sample 5】

See ***bargain/bargain5.in*** and ***bargain/bargain5.ans*** in the user's home path. This sample satisfies the constraints of test $23 \sim 25$.

【Constraints】

For all the tests, it is guaranteed that:

- $1 \leq t \leq 10$;
- $1 \leq n < 10^{10^5}$;
- For each $1 \leq i \leq 9$，$1 \leq v_i \leq 10^5$;
- n consists only of digits $1 \sim 9$.

Test Id	$n <$	$v_i \leq$	Special Properties
1	100	10^5	-
2	10^3		
$3 \sim 6$	10^{18}		
$7 \sim 9$	10^{40}		
10	10^{10^5}		n consists of only one type of digits.
11			n consists of at most two types of digits.
$12, 13$			n consists of at most three types of digits.
$14 \sim 16$	10^{10^3}		$v_1 = v_2 = v_3 = \cdots = v_9$
$17, 18$	10^{10^5}		
$19, 20$	10^{100}	100	-
$21, 22$	10^{10^3}	10^3	
$23 \sim 25$	10^{10^5}	10^5	

Water（water）

【Description】

Little S has a tree with n vertices, rooted at vertex 1. On vertex i ($1 \leq i \leq n$), there is a water cup with an initial temperature of a_i.

Little S, who picked up a water cup and drank water without knowing the temperature and was scalded many times, decided to change all the temperatures of the water in the cups to 0.

Now, Little S can do the following two types of operations an arbitrary number of times (possibly zero):

- Use a heater at vertex i. This will make the temperature of the water cups on all vertices in subtree i increase by 1;
- Or, blow a gust of wind from a **leaf** i. This will make the temperature of the water cups on the route between 1 and i decrease by 1.

Help Little S determine whether he can make the temperature of all the water cups become 0.

【Input Format】

Read from the standard input.

Each test contains multiple test cases.

The first line of the input contains a single integer c —the id of the test. $c = 0$ represents that this is a sample test.

The second line contains a single integer t —the number of test cases.

For each test case:

- The first line contains a single integer n —the number of vertices.
- The second line contains $n - 1$ integers $f_2, f_3, \ldots, f_n$, in which f_i is the parent vertex of vertex i. It is guaranteed that $f_i < i$.
- The third line contains n integers a_i —the initial temperature of the water cup on vertex i.

【Output Format】

Write to the standard output.

For each test case:

- If it is possible to make the temperature of all the water cups 0, print one line, a string `Huoyu`;
- If it is impossible to do so, print one line, a string `Shuiniao`.

【Sample Input 1】

```
0
5
5
1 1 2 3
6 5 2 2 1
5
1 1 2 2
6 5 1 2 1
5
1 1 2 2
4 -1 5 -2 -2
5
1 1 2 2
6 -4 8 -3 -3
5
1 1 2 2
-1 -1 -1 -4 -1
```

【Sample Output 1】

```
Shuiniao
Huoyu
Shuiniao
Shuiniao
Huoyu
```

【Sample Explanation 1】

Let A_u be a heating operation on vertex u, B_u be a wind operation from vertex u and $(S)^k$ be repeating the operations in S for k times.

- In the first, third and fourth test case, it can be proven that Little S can't make the temperature of all the water cups 0;
- In the second test case, a possible operation sequence is $B_3(A_4)^3(B_4)^5B_5$;
- For the fifth test case, a possible operation sequence is $(A_4)^3A_1$.

【Sample 2】

See ***water/water2.in*** and ***water/water2.ans*** in the user's home path.
This sample satisfies the constraints of test 3.

【Sample 3】

See ***water/water3.in*** and ***water/water3.ans*** in the user's home path.
This sample satisfies the constraints of test $7, 8$.

【Sample 4】

See ***water/water4.in*** and ***water/water4.ans*** in the user's home path.
This sample satisfies the constraints of test 12.

【Sample 5】

See ***water/water5.in*** and ***water/water5.ans*** in the user's home path.
This sample satisfies the constraints of test $13, 14$.

【Sample 6】

See ***water/water6.in*** and ***water/water6.ans*** in the user's home path.
This sample satisfies the constraints of test $15 \sim 17$.

【Sample 7】

See ***water/water7.in*** and ***water/water7.ans*** in the user's home path.
This sample satisfies the constraints of test $18 \sim 21$.

【Constraints】

Let $\sum n$ be the sum of n over all test cases in a single test.
For all the tests, it is guaranteed that:

- $1 \le t \le 1\,000$;
- $2 \le n \le 10^5$, $\sum n \le 10^6$;
- For each $2 \le i \le n$, $1 \le f_i < i$;
- For each $1 \le i \le n$, $-10^{12} \le a_i \le 10^{12}$.

Test Id	$n \leq$	$\sum n \leq$	$\lvert a_i \rvert \leq$	Special Properties
1	5	50	5	-
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	3×10^5	10^{12}	A
$15 \sim 17$				B
$18 \sim 21$				C
22, 23	3×10^4	10^5	10^8	-
24, 25	10^5	10^6	10^{12}	

- Property A: For each $2 \leq i \leq n$, $f_i = i - 1$;
- Property B: For each $1 \leq i \leq n$, $a_i \leq \left(\sum_{f_j=i} a_j\right) + 5$, where $f_1 = 0$;
- Property C: The depth of the tree does not exceed 2, where the depth of the tree denotes the maximum number of edges in the simple path from a node to the root.

【Hint】

The input and output files are huge in this problem, please use fast I/O methods.

Anti-palind（anti）

【Description】

We call a string r with length m ***palindrome*** if and only if for each $1 \le i \le m$, $r_i = r_{m+1-i}$ holds.

Given a string s of length n, you have to divide s into several non-empty subsequences such that each of these subsequences is **not** palindrome, and maximize the number of subsequences.

Formally, you have to find a group of sequences $(a_1, a_2, \dots a_k)$ such that:

- For each $1 \le i \le k$, let l_i be the length of a_i, then $l_i \ge 1$ and $1 \le a_{i,1} < a_{i,2} < \dots < a_{i,l_i} \le n$;
- For each $1 \le i \le n$, there is exactly one integer pair (p, q) such that $a_{p,q} = i$;
- For each $1 \le i \le k$, let string $t := s_{a_{i,1}} s_{a_{i,2}} \dots s_{a_{i,l_i}}$, then t is not palindrome.

You have to maximize k, or determine there is no solution.

Specially, if your k is not maximized, you may still get partial scores.

【Input Format】

Read from the standard input.

Each test contains multiple test cases.

The first line of the input contains a single integer c —the id of the test. $c = 0$ represents that this is a sample test.

The second line contains a single integer q —the number of test cases.

For each test case:

- The first line contains a single integer n —the length of the string s.
- The second line contains a string s length n. It is guaranteed that s only contains lowercase English letters.

【Output Format】

Write to the standard output.

For each test case:

- If there is no valid solution, print single string `Shuiniao` on the only line of output.
- Otherwise, you have to:
  - Print a sinle string `Huoyu` on the first line;
  - Print an integer k $(1 \le k \le n)$ —the number of subsequences such that the string s is divided into;
  - For the i-th line of the next k lines $(1 \le i \le k)$:

* First, output an integer l_i ($1 \le l_i \le n$) —the length of the i-th subsequence;
* Next, output l_i integers $a_{i,1}, a_{i,2}, \ldots, a_{i,l_i}$ ($1 \le a_{i,j} \le n$) —the i-th subsequence.

Note that your output should satisfy all the constraints above, otherwise, you will get 0 points in corresponding tests.

【Sample Input 1】

```
0
4
4
kdoi
7
ccccccc
7
sszcdjr
7
abacaca
```

【Sample Output 1】

```
Huoyu
2
2 1 2
2 3 4
Shuiniao
Huoyu
3
3 1 2 3
2 4 5
2 6 7
Huoyu
3
2 1 4
3 2 3 5
2 6 7
```

【Sample Explanation 1】

In the first test case, it is obvious that the output satisfies the constraints. And:

- For the first subsequence, $t =$ `kd` which is not palindrome.
- For the second subsequence, $t =$ `oi` which is not palindrome.

As a result, this is a valid output. It can be proven that for this test case, the maximum possible value of k is 2.

In the second test case, all of subsequences of s are palindrome, so it is obvious that there is no solution.

【Sample 2】

See ***anti/anti2.in*** and ***anti/anti2.ans*** in the user's home path.

This sample has 10 test cases, all satisfying $n = 1\,000$. Among them, the first to third test cases satisfy property A and the fourth to sixth test cases satisfy property B.

【Scoring】

This problem has 20 tests and each worth 5 points.

This problem is judged using ***Special Judge***. Each test case might have multiple solutions and you only need to output one of them.

In each test, your score is the minimum number of points you receive in each test case. For each test case:

- If you incorrectly judged whether there is a solution or print an invalid solution, no points will be given.
- If you correctly judged whether there is a valid solution and print a valid solution:
  - If k is not maximized, 2 points will be given.
  - If k is maximized, 5 points will be given.

【Constraints】

For all the tests, it is guaranteed that:

- $1 \leq q \leq 10$;
- $1 \leq n \leq 10^5$;
- s only contains lowercase English letters.

Test Id	$n \leq$	Special Properties
$1, 2$	5	-
$3 \sim 5$	18	
$6 \sim 8$	1 000	B
$9 \sim 11$		-
$12 \sim 14$	10^5	A
$15 \sim 17$		B
$18 \sim 20$		-

- Property A: It is guaranteed that n is even and each type of letter in s appears no more than $\frac{n}{2}$ times.
- Property B: It is guaranteed that s only consists of `a` or `b`.

【How to Use the Checker】

To help participants test their program, in the `anti` directory in the attachments, `checker.cpp` is provided. Participants can compile their codes and check whether the output is **valid** with it. Note that it is not exactly the same as the checker in final testing. You needn't care its content.

The compile commands are:

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

You can use `checker.cpp` like this:

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

Parameters `<input-file>` and `<output-file>` are respectively the input file and your output file.

If your output is out of range, the checker will report it and exit immediately. Otherwise, the checker will do the following things:

- In the i-th line ($1 \leq i \leq q$), output the detail of the i-th test case.
- In the $(q+1)$-th line, output the summary of this test.

As an example, for the input and output of the first sample, the checker will output the following things to the screen:

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

If the output file is changed to this:

```
Huoyu
2
2 1 2
2 3 4
Huoyu
1
7 1 2 3 4 5 6 7
Huoyu
3
3 1 2 3
2 4 5
2 6 7
Huoyu
3
2 1 4
3 2 3 5
2 6 7
```

Then the checker will output the following things to the screen:

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

Note that `checker.cpp` will only check whether your output is valid, and **will not**:

- Check whether you correctly judge if there's a valid solution.
- Check whether k is maximized.

As an example, if the output file of sample 1 is changed to this:

```
Shuiniao
Shuiniao
Shuiniao
Shuiniao
```

`checker.cpp` will still report `ok` as a result.

Super Show（show）

【Description】

Meguru has prepared a super show!

The stage and the waiting rooms can be seen as a directed acyclic graph (DAG) consisting of n vertices and m edges.

The stage is at vertex 1, and the rest vertices are all waiting rooms. It is guaranteed that each vertex has at least one path to vertex 1. For each waiting room, there is a troupe in it initially.

Meguru can send an ***enter*** command to a waiting room u:

- If the troupe in this room has not come to the stage yet, and there exists a path from u to 1 such that there are no troupe waiting for entrance, then the troupe in waiting room u will come to the stage and subsequently exit. Note that a troupe will **not** return to the waiting room after the exit.
- Otherwise, this command is considered as ***inoperative***.

Meguru has a command sequence $a_1, a_2, \ldots, a_k$ and q queries. In each query, an interval $[l, r]$ is given, and Meguru asks you that, if she sends the enter command to waiting room $a_l, a_{l+1}, \ldots, a_r$ successively, how many troupes are still in the waiting room.

Note that each query is **independent**, i.e., before each query, there is a troupe in each waiting room.

【Input Format】

Read from the standard input.

The first line of the input contains a single integer c —the id of the test. $c = 0$ represents that this is a sample test.

The second line contains four integers n, m, k, and q —the number of vertices, the number of edges, the length of the command sequence, and the number of queries.

Then m lines follow, each containing two integer u and v —there is an edge connecting u and v. It is guaranteed that there are no self-loops or multiple-edges.

The next line contains k integers $a_1, a_2, \ldots, a_k$ —the command sequence.

Then q lines follow, each containing two integers l and r —the interval in the query.

【Output Format】

Write to the standard output.

Print q lines, each line containing a single integer —the answer to the query.

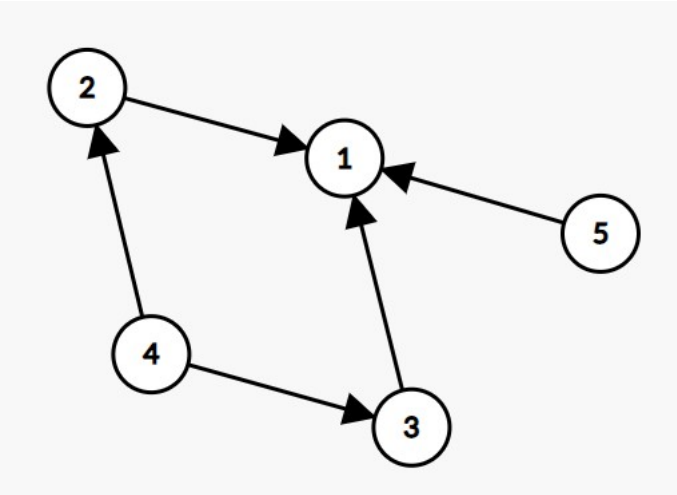
【Sample Input 1】

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

【Sample Output 1】

```
2
0
2
4
```

【Sample Explanation 1】



- In the query $l = 1$, $r = 2$:
  - For the command $a_1 = 2$, 2 comes to the stage by $2 \to 1$;

  - For the command $a_2 = 4$, 4 comes to the stage by $4 \to 2 \to 1$.

  Troupes $3, 5$ are left, so output 2.

- In the query $l = 2$, $r = 3$:
  - For the command $a_2 = 4$, there does not exist a valid path, so this command is inoperative.
  - For the command $a_3 = 4$, there does not exist a valid path, so this command is inoperative.

  Troupes $2, 3, 4, 5$ are left, so output 4.

【Sample 2】

See ***show/show2.in*** and ***show/show2.ans*** in the user's home path.
This sample satisfies the constraints of test $1, 2$.

【Sample 3】

See ***show/show3.in*** and ***show/show3.ans*** in the user's home path.
This sample satisfies the constraints of test $5 \sim 8$.

【Sample 4】

See ***show/show4.in*** and ***show/show4.ans*** in the user's home path.
This sample satisfies the constraints of test $9 \sim 11$.

【Sample 5】

See ***show/show5.in*** and ***show/show5.ans*** in the user's home path.
This sample satisfies the constraints of test $12, 13$.

【Sample 6】

See ***show/show6.in*** and ***show/show6.ans*** in the user's home path.
This sample satisfies the constraints of test $18, 19$.

【Constraints】

For all the tests, it is guaranteed that:

- $1 \le n, k, q \le 2 \times 10^5$;
- $1 \le m \le 4 \times 10^5$;
- $1 \le v < u \le n$, and all (u, v)-s are distinct;

- For each $1 \le i \le k$, $2 \le a_i \le n$;
- For each query, $1 \le l \le r \le k$;
- The input forms a DAG, and each vertex has at least one path to vertex 1.

Test Id	$n, k, q \le$	$m \le$	Special Properties
$1, 2$	300	600	-
$3, 4$	2 000	4 000	A
$5 \sim 8$			-
$9 \sim 11$	2×10^5	4×10^5	A
$12, 13$			BC
$14, 15$			C
$16, 17$			BD
$18, 19$			D
$20 \sim 22$			B
$23 \sim 25$			-

- Property A: The graph is degenerated into an inward-directed tree, i.e. each vertex except vertex 1 has out-degree 1;
- Property B: For each query, $r = k$;
- Property C: For each $1 \le i < j \le k$, $a_i \ne a_j$;
- Property D: The in-degree and out-degree of each vertex does not exceed 30, respectively.