Problem 4 - Rotating... Again?! (100 pts)

Problem Description

Do you recall the game developer, Little Cucumber, who recently launched his game? He invited a magical fairy to give it a try, but to his surprise, the fairy cast a bizarre rotating spell on the game... (See more)

Long story short, you have to maintain a multiset of strings S. At the beginning, S has N strings. Let t_i denote the i-th string in S, $1 \leq i \leq N$. Then, Q operations will be performed on S, where each operation can be one of the following two:

1. `insert(`t_j`)`: insert the string t_j into S.
2. `remove(`t_j`)`: remove one instance of the string t_j from S, it is guaranteed that at least one t_j is in S.

Consider two strings, A and B, of the same length L. A and B are *rotationally identical* if and only if there exists $k, 1 \leq k \leq L$ such that

$$A_1 A_2 \dots A_L = B_k B_{k+1} \dots B_L B_1 \dots B_{k-1} \quad [2]$$

, i.e., if B is rotated by k characters, then A and B are identical.

You are asked to output *the number of string pairs* in S which are *rotationally identical* before the first operation and after each operation.

Input

The first line contains two integers N and Q, the number of strings inside S initially and the number of operations, respectively. Then, the next N lines have t_i in order, $1 \leq i \leq N$, where t_i is a string in S initially. Each of the next Q lines contains an integer P and a string t_j.

- If $P = 1$, it represents an `insert(`t_j`)` operation.
- If $P = 2$, it represents a `remove(`t_j`)` operation.

Output

The output should consist of $Q + 1$ lines. The first line should contain an integer representing the number of rotationally identical string pairs in S initially. Each of the next Q lines should also contain an integer representing the number of rotationally identical string pairs in S after each operation.

[2] $k = 1$ is a special case representing no rotation of B.

Constraints

- $2 \leq N \leq 10^6$
- $0 \leq Q \leq 10^6$
- $1 \leq |t_i| = |t_j| = M \leq 10^6$
- t_i and t_j consists of only lowercase Latin letters.
- $(N + Q)M \leq 10^6$ (the total string length does not exceed 10^6.)

Subtask 1 (10 pts)

- $N, Q, M \leq 50$

Subtask 2 20 pts)

- $Q = 0$

Subtask 3 (20 pts)

- $M = 3$

Subtask 4 (50 pts)

- No other constraints.

Hints

Consider the algorithm you developed for problem 1.2, which calculates the Rabin-Karp hashes for all rotations of a string. Think about how to compare between the hashes of all rotations of two strings.

Sample Testcases

Sample Input 1

```
6 0
ntu
tun
unt
sad
dsa
qqq
```

Sample Output 1

```
4
```

Sample Input 2

```
3 7
bananana
nananaba
cucumber
1 nanabana
1 nabanana
1 anananab
2 cucumber
1 cumbercu
1 bercucum
2 nanabana
```

Sample Output 2

```
1
3
6
10
10
10
11
7
```

Sample Input 3

```
2 3
walnut
walnut
1 walnut
1 walnut
2 walnut
```

Sample Output 3

```
1
3
6
3
```