SAC '22 Code Challenge 3 Junior P5 - Normal Encoding

Time limit: 1.0s
Python: 2.0s

Memory limit: 256M
Python: 512M

To celebrate the 20^{th} anniversary of Home Alone 4, Wesley Alexander Ting-Zhang Leung decided to re-watch it (a 613^{th} time).

While watching the movie, Wesley reminisced,

> I hate this problem.

With the flavour text out of the way, Wesley noted that his Huffman Encoding was wrong; however, Wesley realized that he could still recover the Wesley-ically smallest message.

If two messages have different lengths, the shorter one is Wesley-ically shorter.

If two messages have the same length, the first character that differs between a Wesley-ically smaller string and a larger one will appear earlier in the alphabet for the Wesley-ically smaller string.

Wesley has N pairings of a lowercase letter to a binary string, and the encoded message, M, is made up of the binary strings of characters using the pairings.

Can you help Wesley recover the Wesley-ically smallest message?

Constraints

$1 \leq N \leq 100$

$1 \leq |M| \leq 10^4$

Note that $|M|$ denotes the length of the string M.

M and each binary string will only contain 0 or 1.

The length of each binary string is at most 10 characters.

The binary strings mapped from the same lowercase letter are all the same length.

Note that a letter can map to multiple binary strings.

The data guarantee that there is at least one valid, decoded message.

Subtask 1 [20%]

$N = 1$

The one pairing will have a binary string length of 1.

Subtask 2 [80%]

No additional constraints.

Input Specification

The first line will contain N and $|M|$, the number of pairings of a lowercase letter to a binary string and the length of the encoded string, respectively.

The second line will contain M, the encoded message.

The next N lines will contain a lowercase letter and a binary string, a pairing from that lowercase letter to that binary string.

Output Specification

Output the Wesley-ically smallest message that could be encoded to create S.

Sample Input 1

```
3 5
10101
a 10
b 1
c 0
```

Sample Output 1

```
aab
```

Sample Input 2

```
5 20
10001010001000110001
h 0
h 1
r 0100
i 10001
f 10001
```

Sample Output 2

```
frhff
```