

Problem J. Kitamasa's Counterattack

Input file: *standard input*
 Output file: *standard output*
 Time limit: 2 seconds
 Memory limit: 256 mebibytes

There are n boxes (numbered 1 through n), m keys (numbered 1 through m), and d shops (numbered 1 through d). The key i can be used to open one of the boxes $a_{i,1}, \dots, a_{i,k_i}$. However, once a key is used to open a box, it disappears. Thus, a key can't be used to open multiple boxes. The key i is sold at the shop s_i , and its price is c_i dollars. Akiba wants to buy some keys and open all the boxes. (He can't buy the same key multiple times.)

Kitamasa wants to prevent Akiba from doing this. In order to do it, he can change the prices of some keys before Akiba decides which keys to buy. If he pays b_j dollars, he can increase the prices of all keys sold at the shop j by one dollar. For each shop, he can repeat this any non-negative integer number of times: for example, if he pays $2b_j$ dollars, he can increase the prices of all keys sold at the shop j by two dollars. However, for example when $b_j = 2$, he can't pay 1 dollar and change the prices by 0.5 dollars.

Akiba's objective is to minimize the value (Akiba's payment) $-$ (Kitamasa's payment), and Kitamasa's objective is to maximize it. Compute this value when the two players play optimally. If the answer can be infinitely large, print -1 . It is guaranteed that if Kitamasa does nothing, Akiba can open all boxes.

Input

The first line of input contains three integers n , m , and d ($1 \leq m \leq 1000$, $n \leq 100$, $1 \leq n, d \leq m$).

Then m lines follow, each describing one key. Each line starts with three integers: c_i , the price of the key, s_i , the number of the shop where the key is sold, and k_i , the number of boxes this key can open. Then k_i integers follow: the numbers of these boxes ($1 \leq c_i \leq 1000$, $1 \leq s_i \leq d$, $1 \leq k_i \leq \min(10, n)$, $1 \leq a_{i,j} \leq n$, and if $j \neq k$, $a_{i,j} \neq a_{i,k}$).

Then d lines follow, each containing one integer b_i ($1 \leq b_i \leq 1000$).

Output

Print one integer: the answer to the problem.

Examples

standard input	standard output
3 4 1 2 1 2 1 2 2 1 2 2 3 2 1 2 3 1 3 1 3 1 2 3 5	6
3 4 1 2 1 2 1 2 2 1 2 2 3 2 1 2 3 1 3 1 3 1 2 3 2	-1
2 3 2 3 1 2 1 2 4 1 1 2 5 2 2 1 2 1 2	8