

Yet Another Contest 1 P6 - Alchemy

Time limit: 3.0s **Memory limit:** 512M
Java: 6.0s Java: 1G
Python: 6.0s Python: 1G

After decades of searching, you have finally given up on the eternal fame and glory for discovering the elusive Philosopher's Stone. However, you are still able to make a fortune producing gold, even without the Philosopher's Stone! For the right amount of money, alchemists will perform certain spells, transforming one element into another.

The world consists of N elements labelled from 1 to N , and you currently have a piece of iron (element 1) at your disposal. Your goal is to transform this iron into gold (element N), possibly by transforming the iron into several other elements along the way in order to eventually obtain gold.

There are M different alchemists you can pay, labelled from 1 to M , each of them having their own spells they can perform. Every time you visit a particular alchemist, you can pay them to perform any number of spells. (Here, a visit is counted as a contiguous set of spells performed by the same alchemist, such that no two consecutive visits involve the same alchemist.) The x -th alchemist has an associated value with it, k_x , and is a type t_x alchemist. The three different types of alchemist are detailed below:

- Type 1 alchemists work with k_x different elements. The i -th of these elements is element a_{xi} , which has an associated trading value of b_{xi} . In one spell, they can convert any element from the set into another new element from that set, where the value of the spell is equal to the sum of the trading values of the old and new elements. More specifically, for any $1 \leq i, j \leq k_x$, with $i \neq j$, they can turn element a_{xi} into element a_{xj} for a cost of $b_{xi} + b_{xj}$. The base cost of the visit is equal to the sum of the values of the spells performed.
- Type 2 and type 3 alchemists can perform k_x different spells labelled from 1 to k_x . The i -th spell converts element p_{xi} to element q_{xi} or vice-versa, where the value of the spell is r_{xi} . For type 2 alchemists, the base cost of the visit is equal to the sum of the values of the spells performed. For type 3 alchemists, the base cost of the visit is equal to the maximum of the values of the spells performed.

Furthermore, each time you visit the x -th alchemist, you will have to pay a visiting fee of s_x . The total cost of a visit is hence the sum of the base cost and the visiting fee, and the final cost is equal to the sum of the total costs of each visit.

Note that if you visit an alchemist several times, you will have to pay the visiting fee each time.

Of course, there is no point in producing gold if it costs too much to produce it. What is the minimum possible final cost to transform your piece of iron into gold?

Constraints

$$2 \leq N \leq 2 \times 10^5$$

$$1 \leq M \leq 5 \times 10^5$$

$$1 \leq \sum_{i=1}^M k_i \leq 5 \times 10^5 \text{ (the sum of } k_i \text{ for each alchemist does not exceed } 5 \times 10^5 \text{)}$$

$$1 \leq t_x \leq 3$$

$$1 \leq a_{x_i} \leq N$$

$$1 \leq p_{x_i}, q_{x_i} \leq N, p_{x_i} \neq q_{x_i}$$

$$0 \leq b_{x_i}, r_x, s_x \leq 10^9$$

It is guaranteed that every element is obtainable, starting with element 1.

The elements that type 1 alchemists work with are pairwise distinct.

Subtask 1 [10%]

All alchemists are of type 2. More specifically, $t_x = 2$ for all $1 \leq x \leq M$.

Furthermore, for all $1 \leq x \leq M$, $s_x = 0$.

Subtask 2 [20%]

All alchemists are of type 1 or 2. More specifically, $1 \leq t_x \leq 2$ for all $1 \leq x \leq M$.

Subtask 3 [70%]

No additional constraints.

Input Specification

The first line contains two space separated integers N, M .

The x -th of the following M sets of lines describes the x -th alchemist.

The x -th set of lines begins with a line containing three space separated integers t_x, k_x, s_x , denoting the type, special value and visiting fee respectively of the x -th alchemist.

If $t_x = 1$, then the following line containing k_x space separated integers $a_{x_1}, a_{x_2}, \dots, a_{x_{k_x}}$, corresponding to the set of elements the x -th alchemist works with. This is followed by a line containing k_x space separated integers $b_{x_1}, b_{x_2}, \dots, b_{x_{k_x}}$, corresponding to the associating trading values of the elements that the x -th alchemist works with.

Otherwise, i -th of the following k_x lines contains three space separated integers $p_{x_i}, q_{x_i}, r_{x_i}$, denoting the initial element, new element and value of the i -th spell able to be performed by the x -th alchemist.

Output Specification

On a single line, print the minimum possible final cost of obtaining element N .

Sample Input

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

Sample Output

```
27
```

Explanation

To obtain gold (element 9), you can perform the following:

1. Begin visiting alchemist 1, incurring a visiting fee of 2.
2. Turn your element 1 into element 2, and then turn your element 2 into element 3. The base cost of this visit is $3 + 4 = 7$.
3. Finish visiting alchemist 1.
4. Begin visiting alchemist 2, incurring a visiting fee of 3.
5. Turn your element 3 into element 6. The base cost of this visit is $1 + 4 = 5$.
6. Finish visiting alchemist 2.
7. Begin visiting alchemist 3, incurring a visiting fee of 4.
8. Turn your element 6 into element 7, and then turn your element 7 into element 8. The base cost of this visit is $\max(3, 2) = 3$.
9. Finish visiting alchemist 3.
10. Begin another visit to alchemist 1. Note that you will have to pay the visiting fee of 2 again.
11. Turn your element 8 into element 9. The base cost of this visit is 1.
12. Finish visiting alchemist 1.

The final cost is $(2 + 7) + (3 + 5) + (4 + 3) + (2 + 1) = 27$. It can be shown that this is the minimum possible final cost to obtain gold.