I Good Soup

TIME LIMIT: 10.0S
MEMORY LIMIT: 256MB

Dario loves vegetable soups and he is constantly trying to push their taste to new heights. He already bought the ingredients for today's soup and now is the time for mixing. He starts by adding the first ingredient into the mixer. Then he adds the second one. Then the third, and so on. Each time, the soup gets harder to mix. More specifically, there are M relationships of the form (i, j, a, b). This means that if he adds ingredient j to a soup that already contains a different ingredient i, then the mixer will use a extra units of energy. Similarly, if he adds ingredient i to a soup that already contains ingredient j, then the mixer will use b extra units of energy.

In this problem, assume that the final soup will be the same regardless of the order in which you add the ingredients (not true in real cooking). Since Dario also loves optimization problems, he would like to find the most efficient recipe: the one that uses the least amount of energy to mix everything. Output the minimum amount of energy required.

INPUT

The first line contains two integers, n and m: the number of ingredients and the number of energy relationships ($2 \leq n \leq 24$, $1 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Each of the following m lines contains four integers: i, j, a, and b ($1 \leq i, j \leq n$, $i \neq j$, $1 \leq a, b \leq 10^6$). No pair (i, j) will appear multiple times. The pairs (i, j) and (j, i) cannot appear simultaneously.

OUTPUT

Print one integer, the minimum amount of energy required to mix everything.

SAMPLES

Sample input 1	Sample output 1
3 3 1 2 3 5 1 3 7 3 2 3 5 6	12

Explanation of sample 1.
We can choose to add the three ingredients in the order $3, 1, 2$, which will add to the cost:

- 3 for adding 2 after 1
- 3 for adding 1 after 3
- 6 for adding 2 after 3

The total cost will then be 12, and there is no other order that gives us a smaller cost.