


[Enter](#) | [Register](#)
[HOME](#) [TOP](#) [CONTESTS](#) [GYM](#) [PROBLEMSET](#) [GROUPS](#) [RATING](#) [EDU](#) [API](#) [CALENDAR](#) [HELP](#) **10 YEARS!**
[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [HACKS](#) [ROOM](#) [STANDINGS](#) [CUSTOM INVOCATION](#)

D. Incorrect Flow

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

At the entrance examination for the magistracy of the MSU Cyber-Mechanics Department Sasha got the question about Ford-Fulkerson algorithm. He knew the topic perfectly as he worked with it many times on programming competition. As the task for the question he was given a network with partially build flow that he had to use in order to demonstrate the workflow of the algorithm. He quickly finished to write the text and took a look at the problem only to understand that the given network is incorrect!

Suppose you are given a directed graph $G(V, E)$ with two special nodes s and t called source and sink. We denote as n the number of nodes in the graph, i.e. $n = |V|$ and m stands for the number of directed edges in the graph, i.e. $m = |E|$. For the purpose of this problem we always consider node 1 to be the source and node n to be the sink. In addition, for each edge of the graph e we define the capacity function $c(e)$ and flow function $f(e)$. Function $f(e)$ represents the correct flow if the following conditions are satisfied:

- For each edge $e \in E$ the flow is non-negative and does not exceed capacity $c(e)$, i.e. $0 \leq f(e) \leq c(e)$.
- For each node $v \in V$, that is not source or sink ($v \neq s$ and $v \neq t$) the sum of flows of all edges going in v is equal to the sum of the flows among all edges going out from v . In other words, there is no flow stuck in v .

It was clear that as the exam was prepared last night and there are plenty of mistakes in the tasks. Sasha asked one of the professors to fix the network or give the correct task, but the reply was that the magistrate student should be able to fix the network himself. As the professor doesn't want the task to become easier, he asks Sasha to fix the network in a such way that the total number of changes is minimum possible. Sasha is not allowed to remove edges, add new ones or reverse the direction of existing edges. The only thing he is able to do is to change capacity function $c(e)$ and flow function $f(e)$. Moreover, all the values should remain non-negative integers. There is no requirement on the flow to be maximum in any sense.

Find the minimum possible total change of the functions $f(e)$ and $c(e)$ that Sasha has to make in order to make the flow correct. The total change is defined as the sum of absolute differences, i.e. if new functions are $f^*(e)$ and $c^*(e)$, then the total change is $\sum_{e \in E} |f(e) - f^*(e)| + |c(e) - c^*(e)|$.

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 100$, $0 \leq m \leq 100$) — the number of nodes and edges in the graph respectively. Each of the following m lines contains the description of the edges, consisting of four integers u_i, v_i, c_i and f_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq c_i, f_i \leq 1\,000\,000$) — index of the node the edges starts from, the index of the node the edge goes to, current capacity and flow value.

Node number 1 is the source, and node number n is the sink. It's guaranteed that no edge goes to the source, and no edges starts in the sink.

Given graph contains no self-loops but may contain multiple edges.

Output

Print one integer — the minimum total sum of changes that Sasha has to do in order to get the correct flow description.

Examples

input

Copy

```
2 1
1 2 2 1
```

AIM Tech Round 3 (Div. 1)

Finished

→ Practice?

Want to solve the contest problems after the official contest ends? Just register for practice and you will be able to submit solutions.

[Register for practice](#)

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)

→ Problem tags

flows *2900

No tag edit access

→ Contest materials

- Announcement (en)
- Tutorial (en)

output	Copy
0	

input	Copy
2 1 1 2 1 2	

output	Copy
1	

input	Copy
3 3 1 2 1 1 2 3 2 2 1 3 3 3	

output	Copy
1	

input	Copy
4 2 2 3 1 1 3 2 1 1	

output	Copy
0	

Note

In the first sample, the flow is initially correct. Note, that the flow is not maximum, but this is not required.

In the second sample, the flow value of the only edge is greater than its capacity. There are two ways to fix this: either increase the capacity up to 2 or reduce the flow down to 1.

In the third sample, there is only 1 unit of flow coming to vertex 2, but there are 2 units going out of it. One of the possible solutions is to reduce the value of the flow on the second edge by 1.

In the fourth sample, there is isolated circulation of flow, but this description is correct by definition.

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Sep/11/2020 08:49:11^{UTC+8} (f2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#).

Supported by



ITMO UNIVERSITY