Task 5: Aesthetic (`aesthetic`)

Syrup the Turtle lives in a town with a layout consisting of N locations connected by M roads. Each location is indexed from $1, 2, ..., N$ and each road is indexed from $1, 2, ..., M$. The ith road directly connects locations A_i and B_i, has a length of W_i units, and can be traversed in either direction. These roads and locations are arranged such that it is possible to directly or indirectly travel between any pair of locations, and no two roads share the same endpoints.

Each road has its unique scenery, and the town's inhabitants have long come to a common consensus on how aesthetically pleasing each road is to travel through. The present-day index order of the roads reflects this; the roads are increasingly aesthetic from 1 to M.

Recently, the residents have sought to further refine the appeal of the town's topography. After weighing the many functional and aesthetic considerations of this task, an incredible compromise has been struck - an identical replica of a more aesthetic road is to be constructed within the trail of a strictly less aesthetic one. This action is possible for any pair of roads, and will extend the less aesthetic road by the length of the more aesthetic road. In other words, road j may be replicated into road i if and only if $i < j$, and doing so will change the length of road i to $W_i + W_j$. Now, all that remains is for a vote to be cast for one pair of roads to actually be chosen for this project.

Syrup makes frequent trips between his house in location N and the main square at location 1. He would like to know in advance how long the minimum distance between these two locations could get once the project is completed; your task is to determine the value of this distance.

Input

Your program must read from standard input.

The first line contains two integers, N and M.

M lines will follow. The i^{th} line contains three integers, A_i, B_i, and W_i, describing a single road.

Output

Your program must print to standard output.

The output should contain a single integer on a single line, the maximum possible length of the shortest path between locations 1 and N after the project is enacted.

Implementation Note

As the input lengths for subtasks 3, 4, 5, 6 and 7 may be very large, you are recommended to use C++ with fast input routines to solve this problem. The scientific committee does not have a solution written in Python that can fully solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `Aesthetic.java` and place your main function inside `class Aesthetic`.

Subtasks

The maximum execution time on each instance is 2.0s, and the maximum memory usage on each instance is 1GiB. For all testcases, the input will satisfy the following bounds:

- $3 \leq N \leq 300\,000$
- $2 \leq M \leq 300\,000$
- $1 \leq A_i \neq B_i \leq N$
- $0 \leq W_i \leq 10^9$

Your program will be tested on input instances that satisfy the following restrictions:

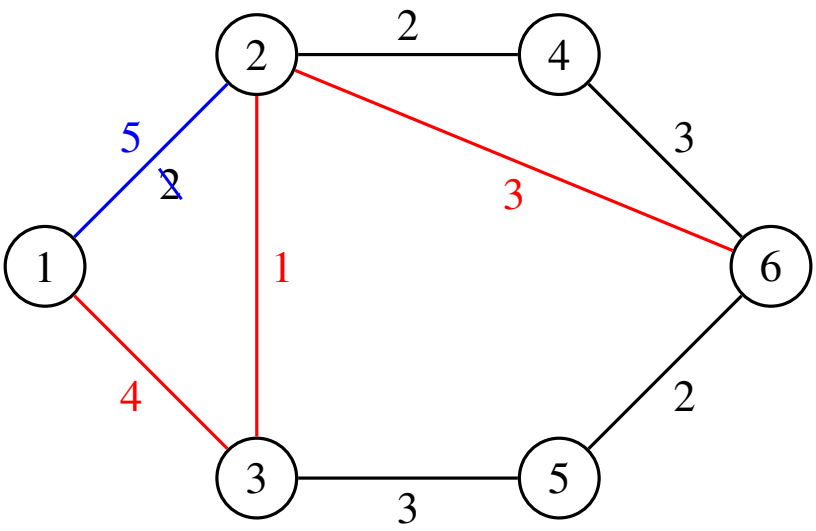
Subtask	Marks	Additional Constraints
1	5	$N, M \leq 100$
2	8	$N, M \leq 2000$
3	7	$M = N - 1$
4	15	$M = N$
5	16	$W_i = 1$
6	22	$0 \leq W_i \leq 10$
7	27	-

Sample Testcase 1

This testcase is valid for subtasks 1, 2, 6, and 7.

Input	Output
6 8 5 6 2 3 1 4 1 2 2 6 2 3 5 3 3 3 2 1 4 6 3 2 4 2	8

Sample Testcase 1 Explanation



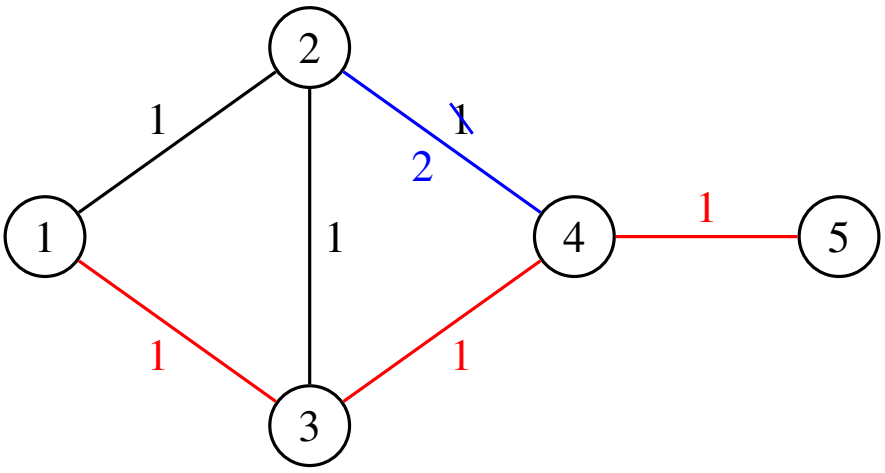
The shortest path between locations 1 and 6 is initially $1 \longrightarrow 2 \longrightarrow 6$, with length 5. If road 3, marked in blue, happens to be extended by any of the more aesthetic roads with length 3, the length of the shortest path could increase to 8 as marked in red.

Sample Testcase 2

This testcase is valid for subtasks 1, 2, 5, 6, and 7.

Input	Output
5 6 1 2 1 4 3 1 2 4 1 3 2 1 1 3 1 4 5 1	3

Sample Testcase 2 Explanation



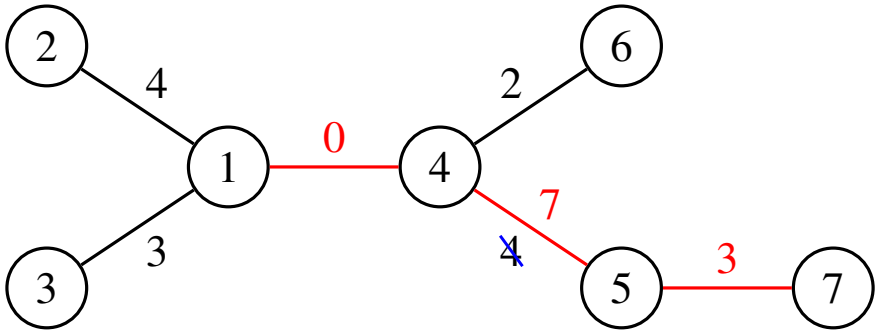
For this testcase, it can be proven that extending a less aesthetic road by a more aesthetic one will never increase the length of the shortest path beyond its original value of 3.

Sample Testcase 3

This testcase is valid for subtasks 1, 2, 3, 6, and 7.

Input	Output
7 6 2 1 4 1 3 3 4 5 4 5 7 3 4 6 2 1 4 0	10

Sample Testcase 3 Explanation



Sample Testcase 4

This testcase is valid for subtasks 1, 2, 4, 6, and 7.

Input	Output
5 5 4 3 3 1 4 4 3 1 3 4 5 2 2 3 1	8

Sample Testcase 4 Explanation