

Happy Feet

Input File: *standard input*

Output File: *standard output*

Forget space flights, undersea submarine voyages and desert treks. Today's frontier of tourism is watching the famous dancing penguins of Antarctica. As the founder of a new company selling hiking tours in the southern-most continent, your killer whale of a business plan is to find the perfect hiking tour of the penguin colonies that will be most enjoyable for the wealthy tourists.

After lengthy discussions with Antarctic penguin experts, you have developed a map of all the human-walkable trails in the region. Each trail leads from one intersection to another intersection and can be walked in either direction. Note that a trail may lead from an intersection back to itself, and there may be multiple trails between two intersections. The experts have given you accurate figures of how many penguins will be seen dancing on each trail. You would like to find a path with the highest total number of penguins seen along the way. This path can start at any intersection and finish at any intersection (including the starting intersection). It may even involve visiting the same intersection multiple times. The only condition is that **each trail in the path must have strictly more penguins than any previous trail**. Tourists exhibit a strange psychological complex requiring their expectations to be continually exceeded. Failure to do so would mean a catastrophic post-trip influx of negative travel reviews.

Given the trail map, write a program to find the largest total number of penguins that can be seen on such a path.

Input

- The first line of input will contain two integers N and M , representing the number of intersections and number of trails respectively.
- The following M lines of input will each contain three integers a_i , b_i , and p_i representing a trail. The trail connects intersections a_i and b_i and has p_i penguins that can be seen dancing on it. Intersections are labelled $1..N$, that is, $1 \leq a_i, b_i \leq N$.

Output

Output should consist of a single integer: the maximum possible number of penguins that can be seen on a path satisfying the described conditions.

Sample Input

```

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

```

Sample Output

```
26
```

Explanation

The optimal path in the sample case above takes the following trails, beginning at intersection 3 and ending at intersection 5.

From	To	Penguins seen
3	1	1
1	4	3
4	3	4
3	4	5
4	5	6
5	5	7

This path has a total of $1 + 3 + 4 + 5 + 6 + 7 = 26$ penguins. No valid path exists with more penguins, hence 26 is the correct answer.

Subtasks & Constraints

In all subtasks, $1 \leq N \leq 1,000$ and each trail will have between 1 and 1,000,000 penguins (inclusive).

- Subtask 1 (30 points): $1 \leq M \leq 1,000$.
- Subtask 2 (40 points): $1 \leq M \leq 1,000,000$ and it is guaranteed that no two trails will have the same number of penguins on them (that is, all trail penguin counts will be unique).
- Subtask 3 (30 points): $1 \leq M \leq 1,000,000$.

You will score the full allocated marks for a subtask if your program returns the correct answer for every test case within that subtask. If your program fails any test case in a subtask, your score will be 0 for that subtask.