

ROI '17 P4 - Travel To Metropolis

Time limit: 2.5s **Memory limit:** 512M

Traveling around the country is never easy, especially when there are no direct connections between cities. A group of tourists want to get to the capital using a railway that connects N cities, numbered from 1 to N . The city from which the group leaves is labeled as 1, and the destination as N .

M train routes are constantly operating on the railway. Each route is determined by a sequence of cities listed in the order in which the train passes them. In each route, for each pair of neighboring cities, the time it takes for a train on this route to travel between these cities is defined. Trains on different routes may pass the same city at different times simultaneously. On the way to the capital, the group can get on and off the train route in any city, not necessarily in the beginning or end (of the route). At the same time, you can get off the train on route i , transfer to a train on route j , perhaps make a few more transfers, and then get back on the train of the same route i .

Tourists have picky requirements for choosing a way to travel to the metropolis.

- First of all, the total time for the trip should be minimal.
- Secondly, among all the methods with the minimal total travelling time, the tourists want to maximize the *quality score*. The *quality score* is defined as the sum of square of the travelling time between two consecutive *transfer* cities in their trip. Check Sample 3 for clarification.

You need to write a program that uses the descriptions of available train routes to determine the minimum time that a group of tourists will have to spend to get from city 1 to city N , as well as the maximum quality of a trip with minimum length.

Input Specification

The first line of input data contains two integers ($2 \leq N \leq 10^6, 1 \leq M \leq 10^6$), the number of cities and the number of routes, respectively. M lines follow, each containing a description of the routes. The description of each route starts with an integer s_i — the number of cities in the i^{th} route ($1 \leq s_i \leq 10^6$) followed by $(2s_i + 1)$ integers, describing the city route and the travel time of the route between consecutive cities of this route in the following order:

$v_{i,1}, t_{i,1}, v_{i,2}, t_{i,2}, v_{i,3}, \dots, t_{i,s_i}, v_{i,s_i+1}$, where $v_{i,j}$ is the i^{th} city in the j^{th} route, and $t_{i,j}$ is the time of travel between j_m and $(j+1)_m$ city ($1 \leq v_{i,j} \leq n, 1 \leq t_{i,j} \leq 1000$).

It is guaranteed that $s_1 + s_2 + \dots + s_m \leq 10^6$. No two cities in a route description match. It is guaranteed that you can use the available routes to get from city 1 to city N .

Output Specification

The output must contain two integers — the minimum total time that you will have to spend travelling from city 1 to city N , and the maximum quality of a trip with minimum length.

Constraints

For all test cases, $2 \leq N \leq 10^6, 1 \leq M \leq 10^6, 1 \leq v_{i,j} \leq n, 1 \leq t_{i,j} \leq 1000, 1 \leq \sum s_i \leq 10^6$.

Subtask	Points	Additional constraints
1	20	$N \leq 1000$
2	20	$N \leq 10^4$
3	20	$N \leq 10^5$
4	20	$N \leq 2 \times 10^5$
5	20	$N \leq 10^6, \sum s_i \leq 10^6$

Sample Input 1

```
2 1
1 1 3 2
```

Sample Output 1

```
3 9
```

Sample Input 2

```
5 2
4 1 3 2 3 3 5 5 10 4
3 4 2 2 1 3 4 1
```

Sample Output 2

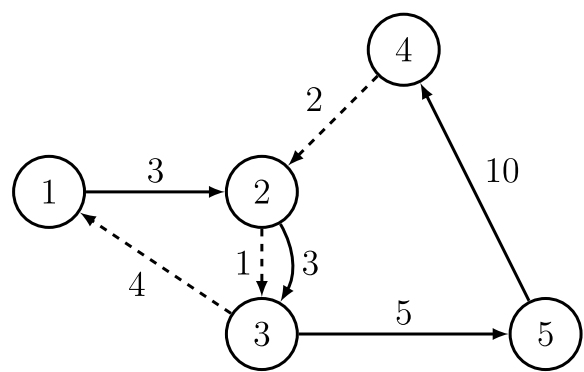
```
9 35
```

Explanation for Sample 2

The following graph shows Sample Case 2. The best path from 1 to 5 is:

- Taking line 1 from city 1 to city 2.
- Transferring to line 2 from city 2 to city 3.
- Transferring to line 1 from city 3 to city 5.

The minimal travelling time is $3 + 1 + 5 = 9$ and the maximal quality score is $3^2 + 1^2 + 5^2 = 35$.



Sample Input 3

```
5 2
3 1 1 2 2 3 3 4
3 2 2 3 3 4 4 5
```

Sample Output 3

```
10 82
```

Explanation for Sample 3

In the third example, you can get from city 1 to city 5 in the shortest possible time by changing trains from route 1 to route 2 in any of the cities 2, 3 or 4. Maximum quality of travel is achieved when transferring in city 2: $1 + 81 = 82$.