

**Euler’s flat in Zurich**

**P84723\_en**

Divuitè Concurs de Programació de la UPC - Final (2020-10-07)

Recently, Gauss has moved to Zurich, and his friend Euler, who used to live in Switzerland, has offered him to move into his empty flat. However, Gauss has found that the flat is a complete mess, so he has to clean every single part of it.

Euler’s flat can be modeled as an undirected graph with  $n$  vertices, one for each spot. The graph has  $m$  edges, each joining two spots that are a distance  $\ell$  apart. In addition,  $k$  spots contain a piece of furniture of weight  $w$ . Note that the  $\ell$ s and the  $w$ s may be different.

To clean every spot  $u$  with furniture, Gauss needs to first clear  $u$ , moving furnitures around in such a way that no two pieces are in the same spot at the same time. The effort to move a piece of weight  $w$  along a distance  $\ell$  is just  $w \cdot \ell$ . After cleaning  $u$ , Gauss will move back all furniture to their original place before cleaning another spot. What is the minimum effort to leave each  $u$  empty? (Do not count the effort to move the furniture back.)

**Input**

Input consists of several cases with only integer numbers. Every case begins with  $n, m$  and  $k$ , followed by  $m$  triples  $u, v, \ell$  denoting an edge between spots  $u$  and  $v$  of length  $\ell$ , followed by  $k$  pairs  $u, w$  meaning that the spot  $u$  contains a piece of furniture of weight  $w$ .

You can assume  $2 \leq n \leq 2 \cdot 10^4$ ,  $1 \leq m \leq 5n$ , and  $0 < k < n$ . Spots are numbered from 0 to  $n - 1$ . All  $\ell$ s and  $w$ s are between 1 and  $10^5$ . The graph is connected, without loops or parallel edges. No spot contains more than one piece of furniture.

**Output**

For each case, and for each spot  $u$  with furniture, in increasing order of  $u$ , print the minimum effort to move the furniture to clean  $u$ . Print a line with 10 dashes after every case.

**Sample input**

```
3 3 2
0 1 5 1 2 9 2 0 15
1 4 2 7

4 3 3
0 1 20000 1 2 60000 2 3 80000
0 50000 1 70000 2 80000

6 7 3
0 3 20 1 3 30 3 2 7 3 4 5
2 5 21 4 5 3 2 4 15
4 9 3 2 2 6
```

**Sample output**

```
1 : 20
2 : 83
-----
0 : 11600000000
1 : 10600000000
2 : 6400000000
-----
2 : 79
3 : 37
4 : 27
-----
```

**Problem information**

Author : Gerard Orriols

Generation : 2024-05-03 00:44:04

© *Jutge.org*, 2006–2024.  
<https://jutge.org>