Collecting Key Servers Statistics

A group of computer servers is connected in a network with undirected tree topology. Some servers are more important and are marked as so-called key servers. From time to time, it is necessary to collect statistics of key servers performance. Statistics collecting is organized as follows.

Servers are labeled by integers. The key server with the smallest label compiles its own performance statistic and sends it as a report to some other key server. Each key server which receives the report compiles its own performance statistic, appends it to the report and sends the report again to another key server which has not received the report yet. The key server which receives the report as last, also appends its own statistic to the report and sends it back to the first key server which initiated the whole process. The report, on its way from one particular key server to another, may pass through other key servers. However, each time the report is processed only by the receiving key server, other servers, including key servers, only mechanically resend the report further.

For each two neighbour servers in the network, the time of message transfer between them is known and it is fixed. The time of statistics collecting is the time elapsed between the moment the first key server sends out its report and the time it finally receives the completed report. The time spent on compiling statistics on each particular key server is negligible and it is not included in the overall statistics collecting time.

The time of statistics collecting may differ depending on particular order in which the key servers send the report from one to another.

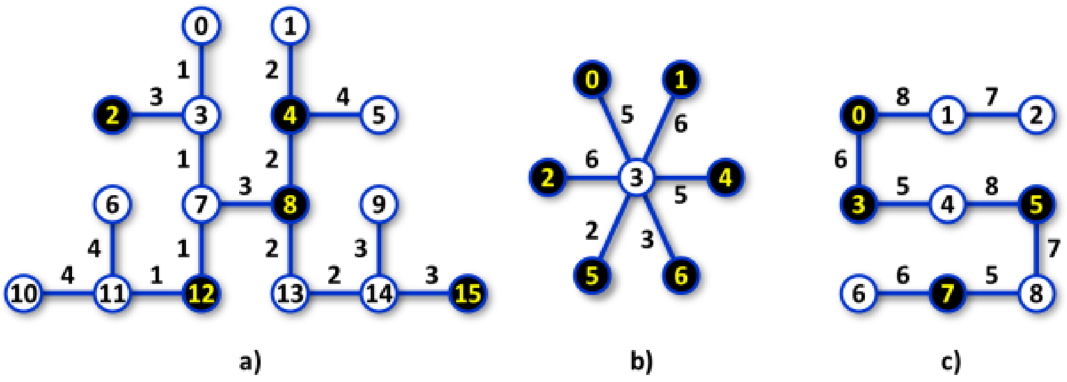

Image 1. Schemes of network connections. The schemes correspond to Examples 1, 2, and 3, below. Key servers are depicted as darker nodes. Message transfer times between two connected servers are written at the corresponding edges between pair of nodes. The minimum time of statistics collecting is 34, 54, 62 in cases a), b), c), respectively. In case a), the message may travel between key servers in order 2 - 12 - 4 - 15 - 8 - 2. In case b), the message may travel between key servers in order 0 - 6 - 1 - 5 - 2 - 4 - 0. In case c), the message may travel between key servers in order 0 - 5 - 7 - 3 - 0. Whenever there are at least three key servers in the network there are often more possible orders of key servers in which statistics collecting may be accomplished and the time spent is minimum possible.

The task

You are given the network topology, list of key servers and the message transfer times between all pairs of neighbour servers. Calculate the minimum possible time of statistics collecting.

Input

The first input line contains two integers N,K separated by space and representing (in this order) the number of all servers in the network and the number of key servers. The servers are labeled by integers 0, 1, ..., N − 1.
The second line contains labels of key servers in arbitrary order, labels are separated by spaces. Next, there are N − 1 lines, each specifies one pair of neighbour servers in the network. Each line contains three integers A, B, T separated by space. Integers A and B are the labels of two neighbour servers, T is the massage transfer time between these two servers. The transfer time is the same in both directions.
It holds, $2 \le N \le 2.5{\times}10^5$, $2 \le K \le 10^4$, $K \le N$. All times of message transfers between neighbour servers are positive and less than 10^3.

Output

Output consists of a single text line with one integer specifying the minimum possible time of statistics collecting.

Example 1

Input

```
16 5
2 12 4 8 15
2 3 3
4 5 5
7 8 3
10 11 4
11 12 1
13 14 2
14 15 3
6 11 4
0 3 1
3 7 1
7 12 1
1 4 2
4 8 2
8 13 2
9 14 3
```

Output

```
34
```

Network scheme in Example 1 is depicted in Image 1a).

Example 2

Input

```
7 6
6 5 4 2 1 0
0 3 5
1 3 6
2 3 6
4 3 5
5 3 2
6 3 3
```

Output

```
54
```

Network scheme in Example 2 is depicted in Image 1b).

Example 3

Input

```
9 4
0 3 5 7
0 1 8
1 2 7
0 3 6
3 4 5
4 5 8
5 8 7
6 7 6
7 8 5
```

Output

```
62
```

Network scheme in Example 3 is depicted in Image 1c).

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

Public data