

# Dreaming Again

**Time limit:** 2.5s    **Memory limit:** 1G

Given a tree with  $N$  nodes, conveniently numbered from 1 to  $N$ , the  $i^{\text{th}}$  edge connects nodes  $u_i$  and  $v_i$  ( $1 \leq u_i, v_i \leq N$ ) with a weight of  $w_i$  ( $-10^6 \leq w_i \leq 10^6$ ). You can remove exactly  $K$  edges from the tree to get  $K + 1$  connected components. After that, you can insert  $K$  new edges with weight 0 to connect all components to form a new tree. The edges you inserted don't have to be the same as the edges you removed. After this whole process, you want to maximize the diameter in this new tree. Can you write a program to find the longest diameter in the new tree?

## Input Specification

The first line contains two integers,  $N$  and  $K$  ( $1 \leq N \leq 3 \times 10^5$ ,  $0 \leq K \leq N$ ), the number of nodes and the number of edges you can remove and insert.

Each of the following  $N$  lines contains three integers,  $u$ ,  $v$  and  $w$  ( $1 \leq u, v \leq N$ ,  $-10^6 \leq w \leq 10^6$ ), an edge connecting nodes  $u$  and  $v$  with weight  $w$ .

## Output Specification

One single integer, the longest diameter in the new tree.

## Constraints

Subtask	Points	Additional constraints
1	10	$K = 0$
2	10	$K = 1$
3	15	$K = 2$
4	25	$K \leq 100$
5	40	No additional constraints.

## Sample Input

```
5 1
1 2 3
2 3 5
2 4 -3
4 5 6
```

# Sample Output

---

14