

Problem G. MIT

Input file: *standard input*
 Output file: *standard output*
 Time limit: 5 seconds
 Memory limit: 998 244 353 bytes

This problem is dedicated to current ICPC champions. From one of 2020 ICPC Champions and, hopefully, 2023 ICPC Champions to 2021 and, hopefully, 2022 ICPC Champions with love.

You are given an edge-weighted tree T on n vertices. Define d_{uv} to be the sum of weights on the only simple path between u and v in T . Consider a full weighted graph G , where the weight of the edge (u, v) is d_{uv} .

For every k between 1 and $\lfloor \frac{n}{2} \rfloor$, calculate the maximum possible weight of a matching of size k in graph G . Recall that a matching of size k is a set of k edges such that no two edges in this set have a common vertex.

Input

The first line contains one integer n ($2 \leq n \leq 100\,000$) — the size of the tree.

The next $n - 1$ lines describe the edges of the tree. The i -th of them contains three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^8$) meaning that there is an edge (u_i, v_i) with weight w_i in T .

It is guaranteed that the given edges form a tree.

Output

Print $\lfloor \frac{n}{2} \rfloor$ integers — maximum weights of matchings of the corresponding sizes in G .

Example

<i>standard input</i>	<i>standard output</i>
7	181 280 287
1 3 99	
2 3 82	
3 4 4	
4 5 43	
5 6 5	
4 7 3	