
Game on a Tree

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

BaoBao is playing a game on a rooted tree with n vertices and $(n - 1)$ weighted edges. At the beginning of the game, a chess piece is placed on the root of the tree, which is vertex 1.

In each step, BaoBao can perform one of the four operations:

1. Move the chess piece along an edge from its current vertex to a child vertex. This operation will cost BaoBao w units of value, where w is the weight of the edge.
2. Jump the chess piece back to vertex 1. This operation is free and won't cost BaoBao any unit of value.
3. Set a "save point" on the current vertex of the chess piece. If a save point has been set on some other vertex before, the save point on the old vertex will be removed (so there will be at most one save point on the tree at the same time). This operation is free and won't cost BaoBao any unit of value.
4. Jump the chess piece back to the save point (the save point must be set before this operation). This operation is free and won't cost BaoBao any unit of value.

The goal of the game is to visit every leaf vertex of the tree using the chess piece. Please help BaoBao calculate the minimum units of value he has to spend to achieve the goal.

Recall that a leaf vertex of a tree is a vertex with no child.

Input

There are multiple test cases. The first line of the input contains an integer T , indicating the number of test cases. For each test case:

The first line contains one integer n ($2 \leq n \leq 200$), indicating the size of the tree.

The following $(n - 1)$ lines each contains three integers u_i , v_i and w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^9$), indicating an edge connecting vertex u_i and vertex v_i with weight w_i in the tree.

It's guaranteed that the given graph is a tree, and the sum of n over all test cases will not exceed 2000.

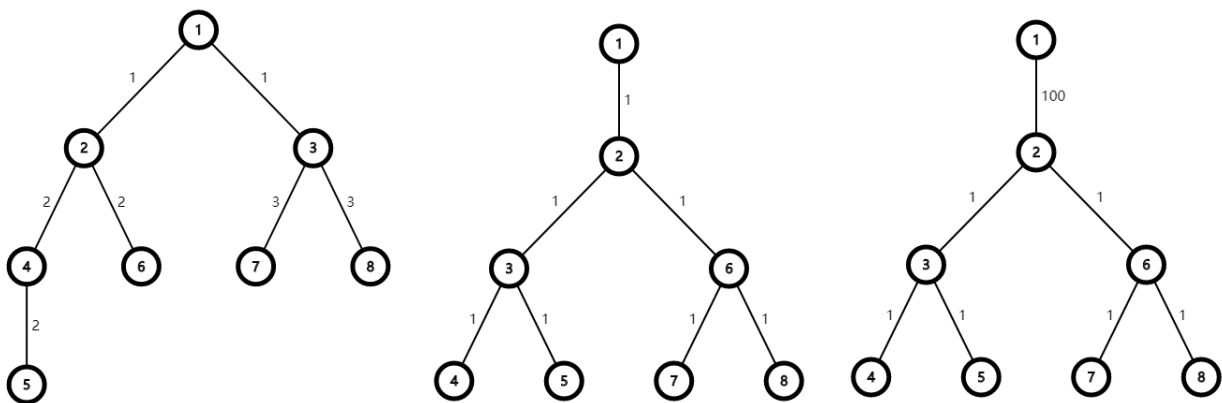
Output

For each test case output one line containing one integer, indicating the minimum units of value BaoBao has to spend to achieve the goal.

Example

standard input	standard output
3	14
8	8
1 2 1	107
3 1 1	
2 4 2	
5 4 2	
6 2 2	
3 7 3	
8 3 3	
8	
1 2 1	
2 3 1	
3 4 1	
3 5 1	
2 6 1	
6 7 1	
6 8 1	
8	
1 2 100	
2 3 1	
3 4 1	
3 5 1	
2 6 1	
6 7 1	
6 8 1	

Note



The trees of the sample test cases are shown above.

For the first sample test case, BaoBao should perform the following operations: go to 2, save on 2, go to 4, go to 5, back to save (2), go to 6, back to root (1), go to 3, save on 3, go to 7, back to save (3), go to 8.

For the second sample test case, BaoBao should perform the following operations: go to 2, go to 3, save on 3, go to 4, back to save (3), go to 5, back to root (1), go to 2, go to 6, save on 6, go to 7, back to save (7), go to 8.

For the third sample test case, BaoBao should perform the following operations: go to 2, save on 2, go to 3, go to 4, back to save (2), go to 3, go to 5, back to save (2), go to 6, save on 6, go to 7, back to save (6), go to 8.