Inventing Test Data

Preparing a problem set is a very hard task. There are always issues with clarity of problem statements, bugs in our solutions, input or output data, and so on. Sometimes, despite our best efforts, these issues are only found during the contest, and this can really spoil it.

To prevent this from happening in the future, we already started to prepare data for IPSC 2009, and we decided to use your help in doing so. Currently we are working on a simple textbook problem: "Given a weighted undirected complete graph, find its minimum spanning tree." (See the Definitions below if you are not sure what a spanning tree is.)

Almost everythig is already prepared for this problem: the problem statement, our solution, and also the desired output data. The only (and quite important) thing left is the input data. But creating it is not as simple as it looks.

The bad thing that can happen is that a graph can have more than one minimum spanning tree. If we used such a graph in the input data, we would have to write a complicated checker. And we are too lazy to do this. Therefore we want to find an input data that avoids such cases.

Moreover, we want the test data to be good. If all the other edges were much more expensive, the minimum spanning tree would be obvious, and many incorrect algorithms would be able to find it. Therefore we want all the edge weights to be as small as possible.

Definitions

A graph is a set of nodes, and a set of links. Each link connects two nodes. Each pair of nodes is connected by at most one link. Each link is assigned a positive integer (its weight). The sum of the weights of all links in a graph is the weight of that graph.

If every two nodes are connected by a link we say that the graph is complete.

A sequence of nodes v_0, …, v_n such that for each i the nodes v_i and v_{i+1} are connected by a link, is called a path.

If every two nodes in a graph are connected by a path, we say that the graph is connected.

If there is exactly one path between any two nodes we say that graph is a tree.

A spanning subgraph of a connected graph G is a connected graph that contains all nodes of G and some (not necessarily all) of its links.

A spanning subgraph T of a graph G is called the minimum spanning tree of G if and only if no other spanning subgraph has a smaller weight.

Note that a given graph can have more than one spanning tree. Also note that a spanning tree is always a tree.

Problem specification

Given a weighted tree T, you are to find the minimum possible weight of a complete graph G such that T is the only minimum spanning tree of G.

Input specification

The first line of the input file contains an integer T specifying the number of test cases. Each test case is preceded by a blank line.

First line of each test case contains an integer N(N <= 15000) – number of nodes in the tree. The nodes are numbered from 1 to N, inclusive. The following N - 1 lines contain a description of the tree. Each of these lines contains three integers a_i, b_i, w_i(1<= a_i, b_i <=N, 1<= w_i <=10000) meaning that node a_i is connected with node b_i by a link with weight w_i.

Output specification

For each test case, the output shall contain one line containing one integer – the minimum possible weight of a complete graph such that the given tree is its unique minimal spanning tree.

Example

Input:

```
2

3
1 2 4
2 3 7

4
1 2 1
1 3 1
1 4 2
```

output:

```
19
12
```

Hint

In the first test case, we have to add a link between nodes 1 and 3 with weight at least 8.

In the second test case, the optimal graph contains the link 2 - 3 with weight 2, and links 2 - 4 and 3 - 4 with weigths 3 each.