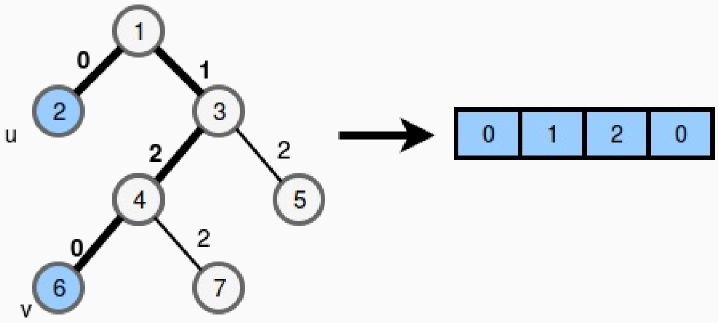
Andy and Lily love playing games with numbers and trees. Today they have a tree consisting of n nodes and $n-1$ edges. Each edge i has an integer weight, w_i.

Before the game starts, Andy chooses an unordered pair of distinct nodes, (u, v), and uses all the edge weights present on the unique path from node u to node v to construct a list of numbers. For example, in the diagram below, Andy constructs a list from the edge weights along the path $(2, 6)$:



Andy then uses this list to play the following game with Lily:

- Two players move in alternating turns, and both players play optimally (meaning they will not make a move that causes them to lose the game if some better, winning move exists).
- Andy always starts the game by removing a single integer from the list.
- During each subsequent move, the current player removes an integer *less than or equal to* the integer removed in the last move.
- The first player to be unable to move loses the game.

For example, if the list of integers is $\{1, 1, 2, 3, 3, 4\}$ and Andy starts the game by removing 3, the list becomes $\{1, 1, 2, 3, 4\}$. Then, in Lily's move, she must remove a remaining integer less than or equal to 3 (i.e., 1, 1, 2, or 3).

The two friends decide to play g games, where each game is in the form of a tree. For each game, calculate the number of unordered pairs of nodes that Andy can choose to ensure he *always* wins the game.

Input Format

The first line contains a single integer, g, denoting the number of games. The subsequent lines describe each game in the following format:

1. The first line contains an integer, n, denoting the number of nodes in the tree.
2. Each line i of the $n-1$ subsequent lines contains three space-separated integers describing the respective values of u_i, v_i, and w_i for the i^{th} edge connecting nodes u_i and v_i with weight w_i.

Constraints

- $1 \leq g \leq 10$
- $1 \leq n \leq 5 \times 10^5$
- $1 \leq u_i, v_i \leq n$
- $0 \leq w_i \leq 10^9$
- Sum of n over all games does not exceed 5×10^5

Scoring

- For 27% of score, the sum of n over all games does not exceed 5000.
- For 100% of score, the sum of n over all games does not exceed 5×10^5.

Output Format

For each game, print an integer on a new line describing the number of unordered pairs Andy can choose to construct a list that allows him to win the game.

Sample Input 0

```
3
5
1 2 2
1 3 1
3 4 1
3 5 2
5
1 2 0
2 3 2
3 4 2
4 5 0
5
1 2 0
1 3 1
3 4 0
3 5 2
```

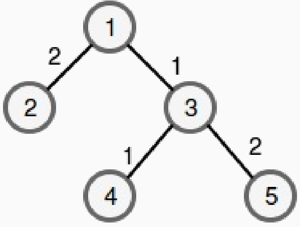
Sample Output 0

```
9
8
10
```

Explanation 0

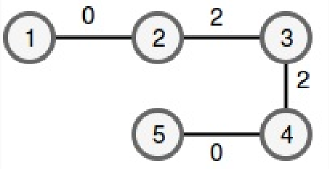
Andy and Lily play the following $g = 3$ games:

1. The first game's tree looks like this:

   There are $\frac{5\cdot(5-1)}{2} = 10$ ways to choose (u, v), and only one such pair causes Andy to lose the game. If he chooses the pair $(1, 4)$, the list is $\{1, 1\}$. Andy removes 1 in his first move, and Lily removes the remaining 1 in the next move; at this point, Andy has no remaining moves and Lily wins. Because Andy will win if he selects any of the other $10 - 1 = 9$ pairs, we print 9 on a new line.

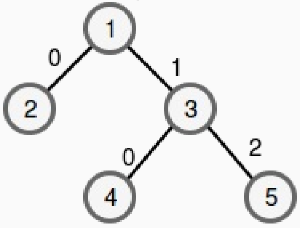
2. The second game's tree looks like this:

   There are $\frac{5\cdot(5-1)}{2} = 10$ ways to choose (u, v), and two pairs that cause Andy to lose the game:

   - If Andy chooses $(2, 4)$, the list is $\{2, 2\}$. Andy removes 2 in his first move, and Lily removes the remaining 2 in the next move; at this point, Andy has no remaining moves and Lily wins.
   - If Andy chooses $(0, 4)$, the list is $\{0, 2, 2, 0\}$. Andy can remove either a 0 or a 2 in the first move, but either way Lily will make an optimal choice in her next move that causes Andy to lose. Andy will win if he selects any of the other $10 - 2 = 8$ pairs, so we print 8 on a new line.

3. The third game's tree looks like this:

   There are $\frac{5\cdot(5-1)}{2} = 10$ ways to choose (u, v), and Andy will win the game regardless of which pair he chooses. Thus, we print 10 on a new line.