Exercise – *Astérix and the Chariot Race*

The condition of the roads in the Roman Empire is disastrous and the Senate is complaining. In order to prove the others wrong and showcase the "excellent" roads, Senator Lactus Bifidus proposes a chariot race across the Italian Peninsula. The race is open to everyone from the known world, including Obélix, who is accompanied by his faithful dog Idéfix and his friend Astérix.

The race track consists of n cities $0, \ldots, n-1$, throughout the Italian Peninsula. Strangely, it is somewhat different from what you might expect of a standard race. Namely, the race starts in *Modicia*[1] (city 0) and ends at one of the several possible *final destinations*. Once the racers have reached one of these final destinations, they are directed towards *Rome* (as it is widely known that all roads lead to Rome) for a celebration and a majestic feast.

The race has *exactly* $n-1$ stages. Each stage can be completed in one direction only, it starts at some city i, and it ends at some city j. Several stages may start at the same city. Furthermore, every city can be reached via a sequence of stages starting from Modicia. The cities from which no stage starts are the *final destinations* of the race.

However, for the race to have the desired effect, the roads in some of the cities need to be repaired, indeed. In order for a city i to be saved from disgrace, either its roads must be repaired, or the roads in at least one city j for which there is a stage of the race between i and j (in either direction). Note that the stages of the race are not considered in this and do not need to be repaired, only the roads in the cities themselves.

Senator Bifidus is in trouble as his budget is limited. Luckily, he knows how much repairing all the roads in every given city costs. Due to the limitations on his budget, he cannot afford to repair the roads in all the cities. Rather, he needs to decide on a subset of cities such that repairing the roads in these cities saves every city from disgrace. In addition, the total reparation costs (sum of the costs for the chosen cities) are to be minimized. Otherwise, exile into Cyrenaica[2] awaits...

Input The first line of the input contains the number $t \leqslant 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that consists of an integer n denoting the number of cities through which the race takes place ($1 \leqslant n \leqslant 10^5$).
- The following $n-1$ lines define the stages of the race. Each line consists of two integers i j, separated by a space, denoting a stage of the race that starts at city i and ends at city j (where $i, j \in \{0, \ldots, n-1\}$).
- The following line defines the costs of repairing the roads in the cities. It consists of n integers $c_0 \ \ldots \ c_{n-1}$, separated by a space ($0 \leqslant c_i \leqslant 2^{14}$, for all $i \in \{0, \ldots, n-1\}$). Here c_i denotes the cost of repairing all the roads in city i, for $i \in \{0, \ldots, n-1\}$.

[1] nowadays known as Monza
[2] nowadays eastern coastal region of Lybia

Output For each test case output a single line that consists of a single integer denoting the minimum total reparation costs so that every city is saved from disgrace.

Points There are four groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 25 points, you may assume that the number of stages it takes to reach any final destination starting from Modicia is always the same. Moreover, you may assume that the costs for repairing the roads in all cities are the same, that is, $c_0 = c_1 = \cdots = c_{n-1}$.
2. For the second group of test sets, worth 25 points, you may assume that the costs for repairing the roads in all cities are the same, that is, $c_0 = c_1 = \cdots = c_{n-1}$.
3. For the third group of test sets, worth 25 points, you may assume that $n \leqslant 10^4$.
4. For the fourth group of test sets, worth 25 points, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3, 4\}$.

Sample Input

```
3
7
0 1
0 2
1 3
1 4
1 5
2 6
17 17 17 17 17 17 17
7
0 1
0 2
1 3
1 4
1 5
3 6
100 17 10 5 3 7 10
6
0 1
1 2
2 3
3 4
4 5
5 100 100 2 2 2
```

Sample Output

```
34
25
9
```