Task: PRZ
Transfers

Olimpiada Informatyczna

XXV OI, Stage II, Day zero (trial). Source file `prz.*` Available memory: 128 MB. *13.02.2018*

Byteasar and his friends want to account for the expenses during their last camping trip together. There are n campers, the i-th of whom currently has x_i bythalers in their bank account and should have y_i bythalers once all the expenses are accounted for.

Bank transfers in Byteotia are rather expensive, but there is a way around this. Everyone can define any number of *buddies* in the banking system. The buddy relation is symmetric, i.e., if A has defined B as their buddy, then A is also B's buddy. To prevent anti-social behavior, no one can be their own buddy. Currently all banks have a peculiar special offer, namely, everyone can issue a free transfer of one bythaler to *each* of their buddies *simultaneously*, and there is no limit on the number of such operations.

The camping friends have defined $n-1$ buddies pairs, forming a network so that every pair of them are connected by a chain of buddies, i.e., it is possible to transfer money between any pair of campers by issuing regular (not free ones from the offer) transfers among buddies only. Now the campers are wondering if it is possible to settle their camping expenses using only the free special transfers. We note that negative account balances are allowed.

Input

The first line of the standard input contains a single integer n ($n \geq 2$) specifying the number of campers. They are numbered from 1 to n.

The second line contains a sequence of n integers $x_1, x_2, \ldots, x_n$ ($0 \leq x_i \leq W$) specifying the current account balances of the campers. The third line contains a sequence of n integers $y_1, y_2, \ldots, y_n$ ($0 \leq y_i \leq W$) specifying the desired final account balances.

Then $n-1$ lines follow, which specify the designated friendships: the i-th such line contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) indicating that the campers no. a_i and b_i are designated friends in the banking system.

Output

A single word should be printed to the first line of the standard output: either `TAK` (Polish for *yes*) or `NIE` (Polish for *no*) depending on whether the free special transfers suffice to settle the accounts. In case of a positive answer, the second line should contain a single integer: the minimum number of special transfers that suffice.

Example

For the input data:

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

the correct result is:

```
TAK
4
```

Explanation for the example: The table below illustrates a way to settle the accounts using four special transfers. Successive lines provide the account balances at each step in the process.

camper no.	1	2	3	4	5
initial balance	4	3	2	1	0
transfer from 2 (to 1 and 4)	5	1	2	2	0
transfer from 5 (to 1)	6	1	2	2	−1
2nd transfer from 2 (to 1 and 4)	7	−1	2	3	−1
transfer from 1 (to 2, 3, and 5)	4	0	3	3	0

Sample grading tests:

`1ocen:` $n = 3$, $x_1 = 1$, $x_2 = x_3 = y_1 = y_2 = y_3 = 0$, answer is `NIE`;

`2ocen:` $n = 1000$, network forms a star, answer is `TAK`;

`3ocen:` $n = 1\,000\,000$, network forms a path, answer is`TAK`.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several unit tests.

Subset	Property	Score
1	$n \leq 10$, $W \leq 5$	20
2	$n \leq 1000$, $W \leq 1\,000\,000$	30
3	$n \leq 1\,000\,000$, $W \leq 1\,000\,000$	50