E. Chemistry in Berland

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Igor is a post-graduate student of chemistry faculty in Berland State University (BerSU). He needs to conduct a complicated experiment to write his thesis, but laboratory of BerSU doesn't contain all the materials required for this experiment.

Fortunately, chemical laws allow material transformations (yes, chemistry in Berland differs from ours). But the rules of transformation are a bit strange.

Berland chemists are aware of n materials, numbered in the order they were discovered. Each material can be transformed into some other material (or vice versa). Formally, for each i $(2 \le i \le n)$ there exist two numbers x_i and k_i that denote a possible transformation: k_i kilograms of material x_i can be transformed into 1 kilogram of material i, and 1 kilogram of material i can be transformed into 1 kilogram of material x_i. Chemical processing equipment in BerSU allows only such transformation that the amount of resulting material is **always an integer number of kilograms**.

For each i $(1 \le i \le n)$ Igor knows that the experiment requires a_i kilograms of material i, and the laboratory contains b_i kilograms of this material. Is it possible to conduct an experiment after transforming some materials (or none)?

Input

The first line contains one integer number n $(1 \le n \le 10^5)$ — the number of materials discovered by Berland chemists.

The second line contains n integer numbers $b_1, b_2... b_n$ $(1 \le b_i \le 10^{12})$ — supplies of BerSU laboratory.

The third line contains n integer numbers $a_1, a_2... a_n$ $(1 \le a_i \le 10^{12})$ — the amounts required for the experiment.

Then $n - 1$ lines follow. j-th of them contains two numbers x_{j+1} and k_{j+1} that denote transformation of $(j+1)$-th material $(1 \le x_{j+1} \le j, 1 \le k_{j+1} \le 10^9)$.

Output

Print `YES` if it is possible to conduct an experiment. Otherwise print `NO`.

Examples

input

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

output

```
YES
```

input

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

output

```
NO
```