Dfs Order 0.5

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Shuishui has a rooted tree of n vertices numbered from 1 to n, where vertex 1 is the root. The value of the i-th vertex is a_i.

Now we start the depth-first search at the root. Because sons of a node can be iterated in arbitrary order, multiple possible depth-first orders exist. We define the value of a depth-first order to be adding up the value of vertices which appear in the depth-first order at an even index. Shuishui wonders among all possible depth-first orders of the given tree, what the maximum value is.

Following is a pseudo-code for the depth-first search on a rooted tree. After calling MAIN(), `dfs_order` will be the depth-first search order.

Algorithm 1 An implementation of depth-first search

```
1: procedure DFS(vertex x)
2:     Append x to the end of dfs_order
3:     for each son y of x do                ▷ Sons can be iterated in arbitrary order.
4:         DFS(y)
5: procedure MAIN()
6:     Let dfs_order be a global variable
7:     dfs_order ← empty list
8:     DFS(1)
```

Following is a pseudo-code for calculating the value of a depth-first order. Calling CALC(`dfs_order`) will return the value of it.

Algorithm 2 An implementation of calculating the value of a depth-first order

```
1: procedure CALC(dfs_order)          ▷ dfs_order is a 1-based array of length n
2:     s ← 0
3:     p ← 2
4:     while p ≤ n do
5:         u ← dfs_order_p
6:         s ← s + a_u
7:         p ← p + 2
8:     return s
```

Input

The input contains multiple testcases.

The first line contains a single integer t ($1 \le t \le 2 \times 10^5$), denoting the number of testcases.

For each testcase:

The first line contains a single integer n ($1 \le n \le 2 \times 10^5$), denoting the number of vertices in the tree.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), denoting the values of these vertices.

Each of the next $n-1$ lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n, u_i \ne v_i$), denoting an edge of the tree.

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each testcase, output a single integer in one line, denoting the answer.

Example

standard input	standard output
3 2 1 2 1 2 4 1 1 222 222 1 2 1 3 2 4 6 1 5 4 6 1 1 6 1 4 5 4 2 1 4 1 3	2 444 15

Note

In the first testcase of the example, the `dfs_order` is unique, which is $[1, 2]$, and its value is $a_2 = 2$. Note that the `dfs_order` is 1-based.

In the second testcase of the example, the `dfs_order` with the maximum value is $[1, 3, 2, 4]$, and its value is $a_3 + a_4 = 444$.