Problem K. Rock Tree

Input file: `standard input`
Output file: `standard output`

Professor Rockdu is interested in tree problems, and recently he has created a new data structure called *Rock Tree*.

Given a constant number k and a tree $T = \{V, E\}$ with V as the node set and E as the edge set, a non-empty set of nodes A is called a *Rock Tree* of T if and only if:

- $A \subseteq V$
- All nodes of A are connected in T, which means for every pair of nodes u and v which are both in A, the nodes in the shortest path between u and v in T are all in A.
- The largest distance over every two nodes in A is not greater than k. The distance between two nodes u and v is defined as the number of nodes (including u and v) in the shortest path between u and v in the tree.

Now Rockdu makes a tree R with n nodes and each node i has a value a_i assigned to it. He wants to find the *Rock Tree* with the maximum sum of node values.

Input

The first line contains a single integer T ($1 \le T \le 100$), denoting the number of test cases.

For each test case, the first line contains two integers n, k ($1 \le n \le 10^5$, $1 \le k \le n$), indicating the number of nodes and the distance limit.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($|a_i| \le 10^4$), indicating the value of nodes.

Each of the following $n - 1$ lines contains two integers u, v ($1 \le u, v \le n$), denoting an edge between u and v. It is guaranteed that these edges form a tree.

It is guaranteed that the sum of n over all test cases won't exceed 10^6, and there are at most 4 test cases with $n > 50000$.

Output

For each test case, output an integer denoting the maximum sum of node values in a single line.

Example

standard input	standard output
2 7 3 3 2 -2 -6 6 3 -7 1 2 2 3 2 4 2 5 5 6 2 7 12 5 0 7 -1 3 -3 10 -1 -1 -5 -1 -4 -9 1 2 1 3 1 4 2 5 4 6 6 7 1 8 8 9 5 10 9 11 9 12	11 20