**Counting on Tree**

You are given a tree consisting of **N** nodes, each node *i* has a value **a[i]** (0 ≤ **a[i]** ≤ 1) associated with it.

We call a set **S** of tree nodes *beautiful* if following conditions are satisfied:

1. **S** is non-empty.
2. **S** is connected. In other words, if nodes **u** and **v** are in **S**, then all nodes lying on the simple path between **u** and **v** should also be presented in **S**.
3. Sum of all **a[u]** (**u** belong to **S**) equals to **K** where **K** is a given integer.

Your task is to count the number of *beautiful* sets. Since the answer may very large, you should print it modulo 109 + 7.

**Input**

The first line contains one integer **T** - the number of test cases. Then **T** test cases follow.

Each test case consists of several lines.

* The first line contains 2 integers **N** and **K**.
* The second line contains **N** integers **a[1]**, **a[2]**, ..., **a[N]**.
* Then the next **N** - 1 line each contain pair of integers **u** and **v** (1 ≤ **u**, **v** ≤ **N**) denoting that there is an edge between **u** and **v**. It is guaranteed that these edges form a tree.

**Output**

For each test, print the answer in one line.

**Constraints**

* Let **SN** be the sum of **N** over all test cases
* 1 ≤ **SN** ≤ 50000
* 1 ≤ **K** ≤ 100
* 10% number of tests in which **SN** ≤ 25
* 30% number of tests in which **SN** ≤ 1000
* 20% number of tests in which **K** ≤ 10

**SAMPLE INPUT**

3

3 1

1 0 1

1 2

1 3

5 2

0 1 0 1 1

1 2

1 3

1 4

2 5

3 0

1 1 0

1 2

2 3

Approach:

Let’s define the state dp[v][j] -> number of beautiful sets in the subtree at node v with value of sum equal to j.