Problem K. DFS

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	1024 mebibytes

You are given a rooted tree of n vertices, and r is the root of the tree. Each vertex x has value a_x.

Let us define the DFS procedure starting from x to find y:

1. Push x on the stack.
2. Check w, the top element of the stack. If $w = y$, the procedure ends. Otherwise, if there is at least one son of w which is not visited, choose one such son with equal probability and push it on the stack.
3. Repeat step 2 until there is no unvisited son.
4. Pop the top element from the stack.
5. Repeat step 2 until the stack is empty.

The procedure is legal if and only if y is in the subtree of x.

Define $f(x, y)$ as the expectation of the minimum value of all vertices which were pushed on the stack during the DFS procedure starting from x to find y.

Now we want to calculate $\sum f(x, y)$ for all legal pairs (x, y). It can be shown that the answer can be expressed as an irreducible fraction $\frac{x}{y}$, where x and y are integers and $y \not\equiv 0 \pmod{998\,244\,353}$. Output the integer equal to $x \cdot y^{-1} \pmod{998\,244\,353}$. In other words, output an integer a such that $0 \le a < 998\,244\,353$ and $a \cdot y \equiv x \pmod{998\,244\,353}$.

Input

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

For each test case, the first line contains two integers n and r ($1 \le n \le 4 \cdot 10^5$, $1 \le r \le n$), denoting the number of vertices in the tree and the root.

The following line contains n integers, the i-th integer of them is $a_i (1 \le a_i \le 10^9)$ denoting the value of vertex i.

Each of the next $n - 1$ lines contains two integers u and v ($1 \le u, v \le n$), denoting an edge of the tree.

It is guaranteed that $\sum n \le 8 \cdot 10^5$. It is also guaranteed that the given graph is indeed a tree.

Output

Output T lines. Each line must contain one integer: the answer to the respective test case.

Example

standard input	standard output
4 1 1 1 3 3 3 3 4 3 1 3 2 6 1 5 2 4 1 3 6 1 2 1 6 2 3 2 4 4 5 5 1 5 4 3 2 1 1 2 1 3 3 4 3 5	1 16 34 499122202