

## 深搜 (dfs)

### 【题目描述】

九条可怜是一个喜欢算法的女孩子，在众多算法中她尤其喜欢深度优先搜索 (DFS)。

有一天，可怜得到了一棵有根树，树根为  $root$ ，树上每个节点  $x$  有一个权值  $a_x$ 。

在一棵树上从  $x$  出发，寻找  $y$  节点，如果使用深度优先搜索，则可描述为以下演算过程：

1. 将递归栈设置为空。
2. 首先将节点  $x$  放入递归栈中。
3. 从递归栈中取出栈顶节点，如果该节点为  $y$ ，则结束演算过程；否则，如果存在未访问的直接子节点，则以均等概率随机选择一个子节点加入递归栈中。
4. 重复步骤 3，直到不存在未访问的直接子节点。
5. 将上一级节点加入递归栈中，重复步骤 3。
6. 重复步骤 5，直至当前一级节点为  $x$ ，演算过程结束。

我们定义  $f(x, y)$  合法当且仅当  $y$  在  $x$  的子树中。它的值为从  $x$  出发，对  $x$  的子树进行深度优先搜索寻找  $y$  期间访问过的所有节点（包括  $x$  和  $y$ ）权值最小值的期望。

九条可怜想知道对于所有合法的点对  $(x, y)$ ， $\sum f(x, y)$  的值。你只需要输出答案对 998 244 353 取模的结果。具体地，如果答案的最简分数表示为  $\frac{a}{b}$ ，输出  $a \times b^{-1} \bmod 998\,244\,353$ 。

### 【输入格式】

从文件 `dfs.in` 中读入数据。

输入包含多组数据，第一行输入数据组数  $T$ 。

对于接下来的每组数据，第一行两个整数  $n, root$ ，分别表示树的大小，树根的编号。

接下来一行  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示树上每个节点的权值。

接下来  $n - 1$  行，每行包含两个整数  $u, v$ ，表示  $u$  和  $v$  之间有一条树边。

### 【输出格式】

输出到文件 `dfs.out` 中。

对于每组数据，输出一行，包含一个整数，代表对于所有合法点对  $(x, y)$ ， $\sum f(x, y)$  对 998 244 353 取模的结果。

### 【样例输入】

见下发文件中的 `dfs_ex1.in` 和 `dfs_ex2.in`。

### 【样例输出】

见下发文件中的 `dfs_ex1.ans` 和 `dfs_ex2.ans`。

**【数据范围与提示】**

对于所有测试点，满足  $1 \leq T \leq 100$ ， $\sum n \leq 800\,000$ ， $1 \leq n \leq 400\,000$ ， $1 \leq root, u, v \leq n$ ， $1 \leq a_i \leq 10^9$ 。

每个测试点的具体限制见下表：

测试点编号	$\sum n \leq$	$n \leq$	特殊限制
1	50	10	无
2 ~ 4	40 000	5 000	无
5 ~ 10	400 000	100 000	无
11	800 000	400 000	树的生成方式随机
12	800 000	400 000	树是一条链
13	800 000	400 000	根的度数为 $n - 1$
14 ~ 20	800 000	400 000	无

对于测试点 11，树的生成方式为：以 1 为根，对于节点  $i \in [2, n]$ ，从  $[1, i - 1]$  中等概率随机选择一个点作为父亲。之后将编号随机重排。