Dynamic Tree Test (Easy)

Time Limit: 3.0s **Memory Limit:** 128M

Today, we'll be practicing modifications on a tree!

Input Specification

The first line contains two integers, N and M, denoting that there are N vertices and M queries.

Then there are N integers on the next line, each containing one number: the initial weight of each vertex.

Then there are $N-1$ lines, each line containing two integers x and y, denoting that there is an edge between x and y in the tree.

Then next line contains the root.

Then there are M lines:

The first number is K.

$K=0$ means change root. The line contains one additional integer x, representing the new root of the tree.

$K=1$ means path modification. K is followed by integers x, y, z. This operation sets z as the vertex weight of all vertices on the path from x to y.

$K=2$ means path increment. K is followed by x, y, z. This operation increments all vertex weights on the path from x to y by z.

$K=3$ means path min. K is followed by x and y, and asks for the min of the weights on the path from x to y.

$K=4$ means path max. K is followed by x and y, and asks for the max of the weights on the path from x to y.

$K=5$ means path sum. K is followed by x and y, and asks for the sum of the weights on the path from x to y.

$K=6$ means change parent. K is followed by x and y. The operations changes the parent of x to y. If y is in the subtree of this operation, do nothing.

$K=7$ means lowest common ancestor (LCA). K is followed by x and y. This operation queries the LCA of x and y.

Output Specification

Print an answer for each query. All answers go on their own lines.

Constraints

$1 \leq N, M \leq 10^5, 1 \leq x, y \leq N$

Subtasks

For 20% of the points, $K \neq 0, K \neq 1, K \neq 2, K \neq 6$.

For 50% of the points, $K \neq 0, K \neq 6$.

All intermediate values can be stored in a signed 32-bit integer.

Sample Input 1

```
5 6
1 3 5 2 10
1 2
1 3
3 4
3 5
3
3 3 2
7 4 1
2 2 5 3
1 3 4 0
4 2 4
5 1 5
```

Sample Output 1

```
1
3
6
17
```

Sample Input 2

```
9 13
100 2 1 3 6 5 4 7 8
1 2
1 3
2 4
2 7
3 6
3 8
3 5
5 9
1
1 1 2 101
2 2 2 101
3 8 5
7 9 4
7 3 8
0 4
7 4 7
0 5
7 1 5
6 9 8
5 6 9
3 8 5
4 4 6
```

Sample Output 2

```
1
1
3
4
5
21
1
202
```