

## Problem H. Mod Graph

Input file: *standard input*  
 Output file: *standard output*  
 Time limit: 1 second  
 Memory limit: 256 mebibytes

You are given an undirected, connected graph with  $n$  vertices. Every vertex  $v$  has a counter that operates modulo  $b_v$ . The initial state of that counter is  $a_v$ . Every time you visit that vertex, it is increased by one (i.e.  $a_v \leftarrow (a_v + 1) \bmod b_v$ ).

You have to process  $q$  queries of the following two types:

- “1  $s$ ”: Suppose that you start at vertex  $s$ , you have to answer whether it is possible to make  $a_v = 0$  for all  $v$ . You are allowed to take an arbitrary walk through  $G$ , i.e. you can use edges and vertices multiple times. Note that starting the walk at  $s$  already counts as visiting  $s$ , i.e. its counter is increased by one initially.
- “2  $v$   $x$ ”: Update  $a_v \leftarrow x$ .

### Input

The first line contains three integers  $n$ ,  $m$ , and  $q$  ( $1 \leq n, q \leq 5 \cdot 10^4$ ,  $0 \leq m \leq 10^5$ ) — the number of vertices, edges, and queries, respectively.

The second line contains  $n$  integers  $a_1, \dots, a_n$ .

The third line contains  $n$  integers  $b_1, \dots, b_n$  ( $0 \leq a_v < b_v \leq 10^9$ ).

The following  $m$  lines contain the edges of the graph. Each of those  $m$  lines contains two integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ,  $u \neq v$ ) describing an edge connecting vertices  $u$  and  $v$ . The given graph is connected.

Finally, there are  $q$  lines describing the queries. They can be in the two formats described above, i.e.

- “1  $s$ ”: ( $1 \leq s \leq n$ )
- “2  $v$   $x$ ”: ( $1 \leq v \leq n$ ,  $0 \leq x < b_v$ )

### Output

For each query of type 1, output YES in one line if it is possible to make  $a_v = 0$  for all  $v$  and NO otherwise.

### Example

<i>standard input</i>	<i>standard output</i>
2 1 4	YES
0 0	NO
3 3	YES
1 2	
1 1	
2 1 1	
1 1	
1 2	

### Note

In the first query, we start at vertex 1 with counter states  $[1, 0]$ .

We move along the walk  $1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2$ .

The counter states change as follows:  $[1, 0] \rightarrow [1, 1] \rightarrow [2, 1] \rightarrow [2, 2] \rightarrow [0, 2] \rightarrow [0, 0]$ .