

## E. Mashmikh's Designed Problem

time limit per test: 4 seconds  
memory limit per test: 512 megabytes  
input: standard input  
output: standard output

After a lot of trying, Mashmikh designed a problem and it's your job to solve it.

You have a tree  $T$  with  $n$  vertices. Each vertex has a unique index from 1 to  $n$ . The root of  $T$  has index 1. For each vertex of this tree  $v$ , you are given a list of its children in a specific order. You must perform three types of query on this tree:

1. find distance (the number of edges in the shortest path) between  $u$  and  $v$ ;
2. given  $v$  and  $h$ , disconnect  $v$  from its father and connect it to its  $h$ -th ancestor; more formally, let's denote the path from  $v$  to the root by  $x_1, x_2, \dots, x_l$  ( $h < l$ ), so that  $x_1 = v$  and  $x_l$  is root; disconnect  $v$  from its father ( $x_2$ ) and connect it to  $x_{h+1}$ ; vertex  $v$  must be added to the end of the child-list of vertex  $x_{h+1}$ ;
3. in the vertex sequence produced by calling function `dfs(root)` find the latest vertex that has distance  $k$  from the root.

The pseudo-code of function `dfs(v)`:

```
// ls[v]: list of children of vertex v
// its i-th element is ls[v][i]
// its size is size(ls[v])
sequence result = empty sequence;
void dfs(vertex now)
{
    add now to end of result;
    for(int i = 1; i <= size(ls[v]); i = i + 1) //loop from i = 1 to i = size(ls[v])
        dfs(ls[v][i]);
}
```

### Input

The first line of input contains two space-separated integers  $n, m$  ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ), the number of vertices of  $T$  and number of queries to perform.

The  $i$ -th of the following  $n$  lines contains an integer  $l_i$  ( $0 \leq l_i \leq n$ ), number of  $i$ -th vertex's children. Then  $l_i$  space-separated integers follow, the  $j$ -th of them is the index of  $j$ -th child of  $i$ -th vertex. Note that the order of these vertices is important.

Each of the following  $m$  lines has one of the following format: " $1 \ v \ u$ ", " $2 \ v \ h$ ", or " $3 \ k$ ". The first number in the line is the type of query to perform according to the problem statement. The next numbers are description of the query.

It's guaranteed that all the queries are correct. For example, in the second-type query  $h$  is at least 2 and at most distance of  $v$  from root. Also in the third-type query there is at least one vertex with distance  $k$  from the root at the time the query is given.

### Output

For each query of the first or third type output one line containing the result of the query.

### Examples

input
4 9
1 2
1 3

1 4  
0  
1 1 4  
2 4 2  
1 3 4  
3 1  
3 2  
2 3 2  
1 1 2  
3 1  
3 2

output

3  
2  
2  
4  
1  
3  
4

input

2 2  
1 2  
0  
1 2 1  
3 1

output

1  
2