Task 4: `Rigged Roads`

Silvermill is on a tight budget lately, and Peanut, its mayor, is intending to demolish some roads to save on maintenance costs. Silvermill can be described as a city with N road intersections and E roads running between them, with the ith road connecting intersections A_i and B_i. Each road intersection is labelled from $1, \ldots, N$ and each road is labelled from $1, \ldots, E$. It is guaranteed it is possible to travel between any pair of road intersections directly or indirectly, and no two roads share the same endpoints.

To facilitate this effort, Peanut has hired you to help assess the maintenance cost of the roads. The task from Peanut is as follows: You need to report a list $W = (W_1, W_2, \ldots, W_E)$ such that W is a permutation of $1, \ldots, E$ and W_i is the cost to keep the ith road.

Peanut will then pick a subset of roads to keep such that:

- All the road intersections remain connected.
- The sum of costs of the kept roads are minimised.

In other words, Peanut will keep the **minimum spanning tree**, based on the weights given by you. Note that the minimum spanning tree is unique since the costs are distinct.

Unknown to Peanut, you have a hidden agenda. You wish to keep a subset R of roads that form a spanning tree. Notice that you can convince Peanut to pick R by carefully choosing W. Your goal is to find the lexicographically smallest[1] permutation W that satisfies the above condition.

In summary, given a subset R of roads that forms a spanning tree, find the lexicographically smallest weight assignment W such that the minimum spanning tree of the city is R.

Input

Your program must read from standard input.

The input starts with a line with two integers N and E.

E lines will follow. The ith line contains two integers, A_i and B_i, describing a single road.

The last line of input will contain $N - 1$ integers, the labels of roads in R, the set of roads you wish to keep.

[1] $(W_1, \ldots, W_E)$ is smaller than $(W'_1, \ldots, W'_E)$ if there exists $1 \leq p \leq E$ such that $W_p < W'_p$ and $W_i = W'_i$ for $i = 1, \ldots, p - 1$.

Output

Your program must print to standard output.

The output should contain E integers on a single line, the lexicographically minimal permutation W that would result in R being selected as the minimum spanning tree.

Subtasks

The maximum execution time on each instance is 2.0s. For all testcases, the input will satisfy the following bounds:

- $1 \leq N, E \leq 3 * 10^5$
- $1 \leq A_i \neq B_i \leq N$
- $1 \leq R_i \leq N$
- It is possible to travel between any two road intersections using only roads in R.

Your program will be tested on input instances that satisfy the following restrictions:

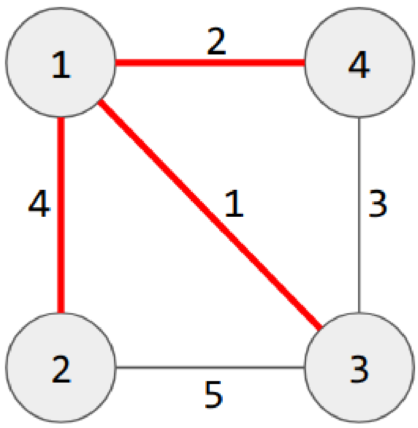
Subtask	Marks	Additional Constraints
1	8	$1 \leq N, E \leq 9$
2	19	$1 \leq N, E \leq 10^3$
3	9	$A_{R_i} = 1$, $B_{R_i} = i + 1$, i.e. R is a star.
4	10	$A_{R_i} = i$, $B_{R_i} = i + 1$, i.e. R is a line.
5	10	$E = N$, $A_i = i$, $B_i = i$ mod N + 1
6	12	$E = N$
7	32	-

Sample Testcase 1

This testcase is valid for subtasks 1, 2, 3, 4 and 7.

Input	Output
4 5 3 4 1 2 2 3 1 3 1 4 2 4 5	3 4 5 1 2

Sample Testcase 1 Explanation



The graph above shows the roads and the road intersections (see input). The numbers on the edges are the weights assigned by you (see output). The highlighted roads are selected by Peanut, which is equivalent to R in the input. As such, the output is correct.

Sample Testcase 2

This testcase is only valid for subtasks 1, 2, 3, 5, 6 and 7.

Input	Output
4 4 1 2 1 4 2 3 3 4 1 3 4	1 4 2 3

Sample Testcase 2 Explanation