

Graph

0.7 s/256 MiB

You are given an undirected graph where each edge has one of two colors: black or red. Your task is to assign a real number to each node so that:

- for each black edge the sum of values at its endpoints is 1;
- for each red edge the sum of values at its endpoints is 2;
- the sum of the absolute values of all assigned numbers is the smallest possible.

Otherwise, if it is not possible, report that there is no feasible assignment of the numbers.

Input

The first line contains two integers N ($1 \leq N \leq 100\,000$) and M ($0 \leq M \leq 200\,000$): the number of nodes and the number of edges, respectively. The nodes are numbered by consecutive integers: $1, 2, \dots, N$.

The next M lines describe the edges. Each line contains three integers a, b and c denoting that there is an edge between nodes a and b ($1 \leq a, b \leq N$) with color c (1 denotes black, 2 denotes red).

Output

If there is a solution, the first line should contain the word “YES” and the second line should contain N space-separated numbers. For each i ($1 \leq i \leq N$), the i -th number should be the number assigned to the node i .

Output should be such that:

- the sum of the numbers at the endpoints of each edge differs from the precise value by less than 10^{-6} ;
- the sum of the absolute values of all assigned numbers differs from the smallest possible by less than 10^{-6} .

If there are several valid solutions, output any of them.

If there is no solution, the only line should contain the word “NO”.

Examples

Input	Output
4 4	YES
1 2 1	0.5 0.5 1.5 -0.5
2 3 2	
1 3 2	
3 4 1	

Input	Output	Comments
2 1	YES	Note that the solution is not unique.
1 2 1	0.3 0.7	

Input	Output
3 2	YES
1 2 2	0 2 0
2 3 2	

Input	Output
3 4	NO
1 2 2	
2 2 1	
2 1 1	
1 2 2	

Grading

Subtasks:

1. (5 points) $N \leq 5$, $M \leq 14$
2. (12 points) $N \leq 100$
3. (17 points) $N \leq 1000$
4. (24 points) $N \leq 10\,000$
5. (42 points) No further constraints

Village

0.7 s/256 MiB

There are N houses in a certain village. A single villager lives in each of the houses. The houses are connected by roads. Each road connects two houses and is exactly 1 kilometer long. From each house it is possible to reach any other using one or several consecutive roads. In total there are $N - 1$ roads in the village.

One day all villagers decided to move to different houses—that is, after moving each house should again have a single villager living in it, but no villager should be living in the same house as before. We would like to know the smallest and the largest possible total lengths in kilometers of the shortest paths between the old and the new houses for all villagers.

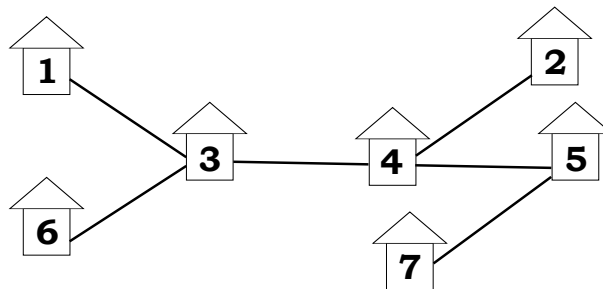


Figure 1: Example village with seven houses

For example, if there are seven houses connected by roads as shown in Figure 1, the smallest total length is 8 km (this can be achieved by moving $1 \rightarrow 6$, $2 \rightarrow 4$, $3 \rightarrow 1$, $4 \rightarrow 2$, $5 \rightarrow 7$, $6 \rightarrow 3$, $7 \rightarrow 5$), but the largest—18 km ($1 \rightarrow 7$, $2 \rightarrow 3$, $3 \rightarrow 4$, $4 \rightarrow 1$, $5 \rightarrow 2$, $6 \rightarrow 5$, $7 \rightarrow 6$).

Write a program that finds the smallest and the largest total lengths of the shortest paths in kilometers and an example assignment of the new houses to the villagers for each of the two cases.

Input

The first line contains an integer N ($1 < N \leq 10^5$). Houses are numbered by consecutive integers $1, 2, \dots, N$.

Then $N - 1$ lines follow that describe the roads. Each line contains two integers a and b ($1 \leq a, b \leq N$, $a \neq b$) denoting that there is a road connecting houses a and b .

Output

In the first line there should be two space-separated integers—the smallest and the largest total lengths of the shortest paths in kilometers.

In the second line describe one valid assignment of the new houses with the smallest total length: N space-separated distinct integers v_1, v_2, \dots, v_N . For each i , v_i is the house number where villager from the house i should move ($v_i \neq i$). If there are several valid assignments, output any of those.

The third line should contain the description of a valid assignment for the largest total length in the same format.

Examples

Input	Output
4	4 8
1 2	2 1 4 3
2 3	4 3 2 1
3 4	

Input	Output
7	8 18
4 2	6 4 1 2 7 3 5
5 7	7 3 4 1 2 5 6
3 4	
6 3	
1 3	
4 5	

Grading

Subtasks:

1. (12 points) $N \leq 10$
2. (38 points) $N \leq 1\,000$
3. (50 points) No further constraints

50% of points will be given if, for each test, the output will contain the correct length and an assignment for any one of the cases (with the smallest or the largest total length). However, the description for both cases should contain N space-separated integers between 1 and N . For the (potentially) incorrect case these may be any values in this range (for example, all 1s).

Viruses

0.7 s/256 MiB

The Committee for Research on Binary Viruses discovered a method of replication for a large family of viruses whose genetic codes are sequences of zeros and ones. Each virus originates from a single gene; for simplicity genes are denoted by integers from 0 to $G - 1$. At each moment in time a virus is a sequence of genes. When mutation occurs, one of the genes from the sequence is replaced by a certain sequence of genes, according to the mutation table. The virus stops mutating when it consists only of genes 0 and 1.

For instance, for the following mutation table:

$$\begin{aligned} 2 &\rightarrow \langle 0\ 1 \rangle \\ 3 &\rightarrow \langle 2\ 0\ 0 \rangle \\ 3 &\rightarrow \langle 1\ 3 \rangle \\ 4 &\rightarrow \langle 0\ 3\ 1\ 2 \rangle \\ 5 &\rightarrow \langle 2\ 1 \rangle \\ 5 &\rightarrow \langle 5 \rangle \end{aligned}$$

a virus that initially consisted of a single gene 4, could have mutated as follows:

$$\langle 4 \rangle \rightarrow \langle \underline{0\ 3\ 1\ 2} \rangle \rightarrow \langle 0\ \underline{2\ 0\ 0}\ 1\ 2 \rangle \rightarrow \langle 0\ \underline{0\ 1}\ 0\ 0\ 1\ 2 \rangle \rightarrow \langle 0\ 0\ 1\ 0\ 0\ 1\ \underline{0\ 1} \rangle$$

or in another way:

$$\langle 4 \rangle \rightarrow \langle \underline{0\ 3\ 1\ 2} \rangle \rightarrow \langle 0\ \underline{1\ 3}\ 1\ 2 \rangle \rightarrow \langle 0\ 1\ 3\ 1\ \underline{0\ 1} \rangle \rightarrow \langle 0\ 1\ \underline{2\ 0\ 0}\ 1\ 0\ 1 \rangle \rightarrow \langle 0\ 1\ \underline{0\ 1}\ 0\ 0\ 1\ 0\ 1 \rangle$$

Viruses are detected by antibodies that identify the presence of specific continuous fragments of zeros and ones in the viruses' codes. For example, an antibody reacting to a fragment $\langle 0\ 0\ 1\ 0\ 0 \rangle$ will detect a virus $\langle 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1 \rangle$, but it will not detect a virus $\langle 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1 \rangle$.

For each gene from 2 to $G - 1$, the scientists are wondering whether a given set of antibodies is enough to detect all viruses that can emerge through mutations from this gene. If not, they want to know the length of the shortest virus that cannot be detected.

It may happen that sometimes scientists don't have any antibodies. Then of course no virus can be detected, so the scientists are only interested in the length of the shortest possible virus that can emerge from the gene mutations.

Input

The first line of the input will contain three integers G , N and M ($G > 2$, $N \geq G - 2$, $M \geq 0$) specifying the number of genes, the number of rows in the mutation table, and the number of antibodies.

The following N lines contain descriptions of rows of the mutation table; each line begins with two integers a and k ($2 \leq a < G$, $k \geq 1$), followed by a sequence of k integers b_1, b_2, \dots, b_k ($0 \leq b_i < G$), that encode the row

$$a \rightarrow \langle b_1\ b_2\ \dots\ b_k \rangle$$

The sum of all values k does not exceed 100. **Every integer from 2 to $G - 1$ appears in the table as a at least once.**

The next M lines contain descriptions of the antibodies; each such line begins with an integer ℓ ($\ell \geq 1$), followed by a sequence of ℓ integers c_1, c_2, \dots, c_ℓ ($0 \leq c_i \leq 1$), describing the antibody. The sum of all values ℓ does not exceed 50.

Output

Your program needs to output exactly $G - 2$ lines, containing the answers for the subsequent genes from 2 to $G - 1$.

If all viruses that can mutate from this single gene can be detected by the given set of antibodies, you need to print the word “YES”. You also need to print this if there are no viruses that could originate from this gene (it can happen when the sequences never stop mutating).

Otherwise you need to print the word “NO”, followed by an integer denoting the minimal length of the undetectable virus. You can assume that for all the prepared input data this value will be smaller than 2^{63} .

Example

Input	Output
6 6 2	NO 2
2 2 0 1	NO 4
3 3 2 0 0	NO 9
3 2 1 3	YES
4 4 0 3 1 2	
5 2 2 1	
5 1 5	
2 1 1	
5 0 0 1 0 0	

Grading

Subtasks:

1. (11 points) No antibodies ($M = 0$)
2. (14 points) $N = G - 2$
3. (25 points) One antibody ($M = 1$)
4. (32 points) The sum of all values ℓ does not exceed 10
5. (18 points) No further constraints