B. Ovenmasters (ovenmasters)

Time limit: 2 seconds
Memory limit: 1024 MiB

You are a reporter at the "Excellent Glutenous Ovenmasters of Italy", an event where the best N pizza bakers of Italy just competed to determine who makes the best pizza. Each baker baked one pizza, and the pizzas were then ranked by a jury. Each pizza received a distinct rank from 0 (best) to $N-1$ (worst). Each baker then received the same rank as their pizza.

After the competition, it is time to eat the pizzas at the pizza gala. All the bakers will attend the event, and everyone will bring their own pizza to the gala. The bakers arrive one by one in some order (not necessarily by rank). At the gala, there are $M \leq N$ tables, numbered from 0 to $M-1$. The first M bakers who arrive place their pizzas on these tables, from 0 to $M-1$ in the order of arrival. Each of the remaining $N-M$ bakers would like to eat a pizza better than theirs, but not too good, so they do not feel bad about themselves. Each time a baker arrives, they choose the available pizza with the worst rank that is still better than theirs. They sit down at the corresponding table to eat their entire chosen pizza. Finally, they leave their own pizza behind on the same table for another baker to potentially eat afterwards. If no suitable pizza exists for an arriving baker (because all tables have pizzas ranked worse than their own), the baker leaves frustrated and takes their pizza with them.

The following example shows a gala with $M = 2$ tables and bakers arriving in the following sequence of ranks: $1, 0, 3, 5, 4, 2$. This gala corresponds to the first example input and output.

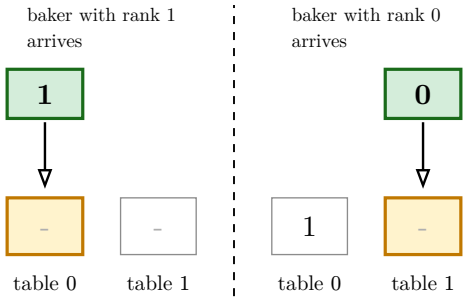

Figure 1: The first $M = 2$ bakers put their pizzas onto the empty tables (0, 1) in order of arrival.

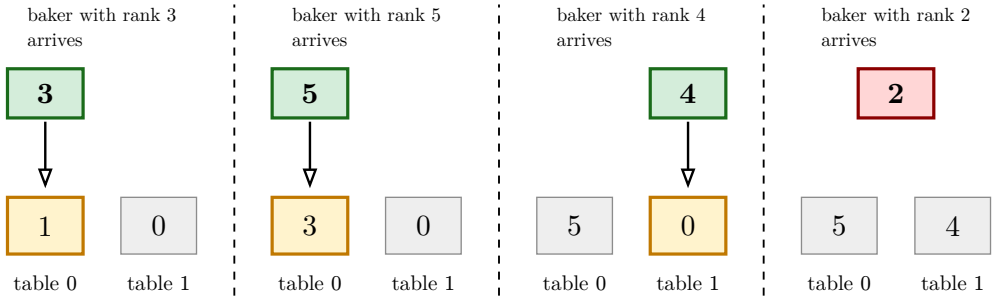

Figure 2: Once all tables are occupied, each arriving baker goes to the table with the worst pizza that is still better than theirs (shown by the arrow), eats that pizza, and leaves their own. If no better pizza exists, the baker leaves frustrated (no arrow).

In your article, you want to report on the order in which the bakers arrived to the pizza gala. Unfortunately, you were too distracted by all the tasty pizzas and forgot to note the order in which

the bakers arrived. Luckily, on each table, you can find a stack of the trays of pizzas that were served at this table in the order in which the pizzas were served.

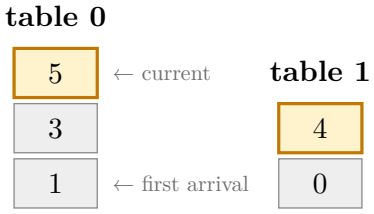

Figure 3: Tray stacks corresponding to the first example. Each stack lists the bakers who were at that table in arrival order, bottom (first) to top (most recent). The highlighted tray has the pizza that was left there at the end of the gala.

You want to use this information to reconstruct the order in which the bakers arrived. You are aware that there might have been several possible orders, so, for full score, you want to report the lexicographically smallest valid order.[1]

Input

The first line contains two integers N and M, the number of bakers and the number of tables.

Then M lines follow, each describing a tray stack on a table. Line i starts with an integer T_i, the number of trays on table i, followed by T_i integers $b_{i,j}$ denoting the rank of the jth pizza that was served at table i.

Output

Output `NO` if there is no possible order satisfying the constraints. Output `YES` if there is a possible order. In this case, output a second line containing N integers $a_0, a_1, ..., a_{N-1}$, the ranks of the bakers in arrival order. If multiple such permutations exist, you should output the lexicographically smallest of them. Note that partially correct answers may still score some points, as explained in the Scoring section.

Constraints

- $1 \leq M \leq N \leq 300\,000$.
- $0 \leq b_{i,j} \leq N - 1$.
- All the $b_{i,j}$ are distinct.
- $1 \leq T_i \leq N$.

Scoring

Your program will be tested on several test cases grouped into subtasks. To obtain the score for a subtask, you must correctly solve all the tests it contains.

⇨ Solutions with only a correct first line (`YES` vs `NO`) will score 20%. Solutions with a correct first line (`YES` vs `NO`) and **any valid** order, not necessarily the lexicographically smallest, when the answer is `YES` will score an additional 20%. To score the remaining 60% you must output the lexicographically smallest valid order when the first line is `YES`.

- **Subtask 0 [0 points]**: Examples.
- **Subtask 1 [20 points]**: $M = 1$.
- **Subtask 2 [10 points]**: $M = 2$, $N \leq 200$, and the sum of all T_i is N (in other words, no baker walks away frustrated).

[1] A sequence $a_0, a_1, \ldots, a_{n-1}$ is lexicographically smaller than a sequence $b_0, b_1, \ldots, b_{n-1}$ if there exists an index $0 \leq t < n$ such that $a_i = b_i$ for all $i < t$ and $a_t < b_t$.

- **Subtask 3 [20 points]**: $M \leq N \leq 200$, and the sum of all T_i is N (in other words, no baker walks away frustrated).
- **Subtask 4 [20 points]**: $M \leq 10$.
- **Subtask 5 [30 points]**: No additional constraints.

Examples

stdin	stdout
6 2 3 1 3 5 2 0 4	YES 1 0 3 5 4 2
6 2 3 1 3 4 2 0 2	NO
4 2 2 0 3 2 1 2	NO
3 1 2 0 2	YES 0 2 1
8 1 8 7 6 5 4 3 2 1 0	NO
12 4 3 2 3 4 1 5 1 6 5 7 8 9 10 11	YES 2 5 6 7 0 1 3 4 8 9 10 11

Explanation

The first example input and output corresponds to the figures shown in the problem statement. In particular, the order in which the bakers arrive to the gala in Figures 1 and 2 is the lexicographically smallest valid arrival order $1, 0, 3, 5, 4, 2$.

In the second example, the tray stacks are inconsistent, since there is no arrival order in which the baker with rank 5 would leave frustrated. Thus, the answer is NO.

In the third, and fifth examples, the tray stacks are also inconsistent (no arrival order can produce them), so the answer is NO.

In the fourth example ($N = 3$, $M = 1$) only one arrival order is possible, namely $0, 2, 1$.

In the sixth example ($N = 12$, $M = 4$) note that the numbers 0 and 1 do not appear among the values $b_{i,j}$. This means that at some point during the gala each of the bakers 0 and 1 walked away frustrated. The example output shows the lexicographically smallest valid arrival order. Other valid arrival orders exist; for instance $2, 5, 6, 7, 8, 1, 3, 4, 9, 10, 11, 0$. Outputting `YES` followed by an alternative valid order such as this one (instead of the lexicographically smallest one) would be considered partially correct for 40% of the score.