

B. 披萨大师 (ovenmasters)

时间限制: 2 秒

空间限制: 1024 MiB

你是一位来自“意大利卓越披萨大师赛”的记者，意大利最顶尖的 N 位披萨师傅刚刚在这里比拼，决出谁才是最好的披萨大师。每位师傅烤了一个披萨，随后由评委根据披萨进行排名。每块披萨都获得了一个从 0（最好）到 $N - 1$ （最差）的唯一排名。每位师傅也获得了与他们披萨相同的排名。

比赛结束后，是披萨盛宴的用餐时间了。所有师傅都会参加，并且每人都带着自己的披萨来到盛宴。师傅们按照某种顺序（不一定是排名顺序）依次到达。盛宴现场有 $M \leq N$ 张桌子，编号从 0 到 $M - 1$ 。前 M 位到达的师傅按到达顺序将披萨放在编号为 0 到 $M - 1$ 的桌子上。剩下的 $N - M$ 位师傅想吃一块比自己做的更好的披萨，但又不能好得太离谱，这样他们才不会感到自卑。每次有师傅到达，他们会从桌面上的披萨选择排名比自己好但最差的那块披萨。他们会在被选择披萨的桌子旁坐下，吃掉选中的整个披萨。最后，他们把自己做的披萨留在桌子上，供后来的师傅（可能）享用。如果对于某位到达的师傅没有合适的披萨（因为所有桌上的披萨排名都比自己的差），这位师傅就会沮丧地离开，并带走自己的披萨（即不留下自己的披萨）。

下面的样例展示了一个拥有 $M = 2$ 张桌子的盛宴，师傅们按以下排名顺序到达：1, 0, 3, 5, 4, 2。这个盛宴对应于第一个样例输入和输出。

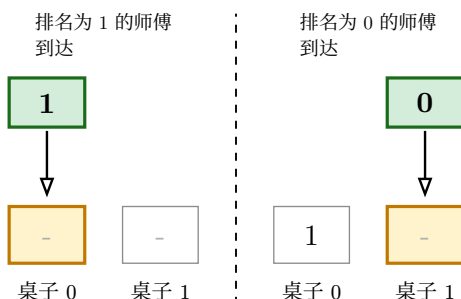


图 1 前 $M = 2$ 位到达的师傅按到达顺序将披萨放在空桌子 (0, 1) 上。

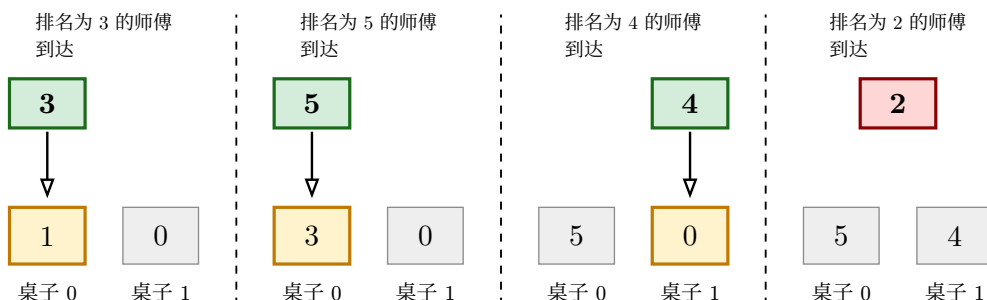


图 2 一旦所有桌子都被占用，每位到达的师傅就会走到桌子上放着（在该条件下）比自己好但排名最差的那块披萨的桌旁（箭头所示），吃掉那块披萨，并留下自己的。如果没有更好的披萨，师傅就会沮丧地离开（无箭头）。

在你的文章中，你想报道师傅们到达披萨盛宴的顺序。可惜，你因为沉迷于各种美味的披萨，忘记记录他们到达的顺序了。幸运的是，在每张桌子上，你可以找到一叠托盘，上面按上菜顺序记录了这张桌子被服务过的披萨。



图 3 对应第一个样例的托盘堆。每堆按到达顺序（从下到上，下为先到）列出了在该桌子就餐的师傅。高亮显示的托盘是盛宴结束时留在桌子上的披萨。

你想利用这些信息还原师傅们到达的顺序。你意识到可能有多种可能的顺序，因此，为了获得满分，你需要报告字典序最小的那个合法顺序。¹

输入

第一行包含两个整数 N 和 M ，分别代表师傅的人数和桌子的数量。

接着有 M 行，每行描述一张桌子上的托盘堆。第 i 行以一个整数 T_i 开头，表示桌子 i 上的托盘数量，后面跟着 T_i 个整数 $b_{i,j}$ ，表示在该桌子被服务过的第 j 块披萨的排名。

输出

如果没有满足条件的可能顺序，输出 NO。如果存在可能的顺序，输出 YES。在这种情况下，输出第二行，包含 N 个整数 a_0, a_1, \dots, a_{N-1} ，表示师傅按到达顺序的排名。如果存在多个这样的排列，你应该输出字典序最小的那一个。注意，部分正确的答案仍可能获得分数，详见评分部分。

约束条件

- $1 \leq M \leq N \leq 300\,000$ 。
- $0 \leq b_{i,j} \leq N - 1$ 。
- 所有的 $b_{i,j}$ 都是唯一的。
- $1 \leq T_i \leq N$ 。

评分方式

你的程序将在分成若干子任务的测试数据上进行测试。要获得某个子任务的分数，你必须正确解出该子任务中所有的测试数据。

⇒ 仅正确回答第一行（YES 或 NO）的解法得 20% 分。正确回答第一行并给出 **任意合法** 顺序（当答案为 YES 时）的解法，额外得 20% 分。要获得剩余 60% 的分数，必须在第一行为 YES 时输出字典序最小的合法顺序。

- 子任务 0 [0 分]: 样例。
- 子任务 1 [20 分]: $M = 1$ 。
- 子任务 2 [10 分]: $M = 2$, $N \leq 200$, 且所有 T_i 之和为 N （换句话说，没有师傅沮丧地离开）。
- 子任务 3 [20 分]: $M \leq N \leq 200$, 且所有 T_i 之和为 N （换句话说，没有师傅沮丧地离开）。
- 子任务 4 [20 分]: $M \leq 10$ 。
- 子任务 5 [30 分]: 没有额外的约束条件。

¹序列 a_0, a_1, \dots, a_{N-1} 在字典序上小于序列 b_0, b_1, \dots, b_{N-1} ，如果存在一个索引 $0 \leq t < n$ ，使得对于所有 $i < t$ ，都有 $a_i = b_i$ 且 $a_t < b_t$ 。

样例

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

样例解释

第一个样例输入和输出对应于题目描述中的图片。特别是，图 1 和图 2 中师傅到达盛宴的顺序是字典序最小的合法到达顺序：1, 0, 3, 5, 4, 2。

在第二个样例中，托盘堆是不合理的，因为不存在一种到达顺序使得排名为 5 的师傅会沮丧地离开。因此，答案是 NO。

在第三和第五个样例中，托盘堆也是不合理的（没有任何到达顺序能产生它们），所以答案是 NO。

在第四个样例中 ($N = 3$, $M = 1$)，只有一种可能的到达顺序，即 0, 2, 1。

在第六个样例 ($N = 12$, $M = 4$) 中，注意数字 0 和 1 没有出现在值 $b_{i,j}$ 中。这意味着在盛宴期间的某个时刻，师傅 0 和 1 都沮丧地离开了。样例输出显示了字典序最小的合法到达顺序。当然还存在其他合法的到达顺序；例如 2, 5, 6, 7, 8, 1, 3, 4, 9, 10, 11, 0。输出 YES 并跟随一个合法的替代顺序（而不是字典序最小的那个）将被视为部分正确，得分为 40%。