

## B. Ofenmeister (ovenmasters)

Zeitlimit: 2 Sekunden

Speicherlimit: 1024 MiB

Du bist Reporter bei den „Excellent Glutenous Ovenmasters of Italy“, einer Veranstaltung, bei der die besten  $N$  Pizzabäcker Italiens gerade gegeneinander angetreten sind, um den besten Pizzabäcker zu ermitteln. Jeder Bäcker hat eine Pizza gebacken und die Pizzen wurden von einer Jury bewertet. Jede Pizza erhielt einen eindeutigen Rang von 0 (beste) bis  $N - 1$  (schlechteste). Jeder Bäcker erhielt daraufhin denselben Rang wie seine Pizza.

Nach dem Wettbewerb ist es Zeit, die Pizzen auf der Pizza-Gala zu essen. Alle Bäcker werden an der Veranstaltung teilnehmen und jeder bringt seine eigene Pizza mit zur Gala. Die Bäcker kommen nacheinander in einer beliebigen Reihenfolge an (nicht unbedingt nach Rang sortiert). Bei der Gala gibt es  $M \leq N$  Tische, die von 0 bis  $M - 1$  nummeriert sind. Die ersten  $M$  Bäcker, die ankommen, platzieren ihre Pizzen in der Reihenfolge ihrer Ankunft auf diesen Tischen, von 0 bis  $M - 1$ . Jeder der verbleibenden  $N - M$  Bäcker möchte gerne eine Pizza essen, die besser ist als seine eigene, aber nicht **zu** gut, damit er sich nicht schlecht fühlt. Jedes Mal, wenn ein Bäcker ankommt, wählt er die verfügbare Pizza mit dem schlechtesten Rang, die immer noch besser ist als seine eigene. Er setzt sich an den entsprechenden Tisch, um seine gesamte gewählte Pizza zu essen. Schließlich hinterlässt er seine eigene Pizza auf demselben Tisch, damit ein anderer Bäcker sie später eventuell essen kann. Wenn für einen ankommenden Bäcker keine passende Pizza existiert (weil auf allen Tischen nur Pizzen liegen, die schlechter sind als seine eigene), geht der Bäcker frustriert und nimmt seine eigene Pizza mit.

Das folgende Beispiel zeigt eine Gala mit  $M = 2$  Tischen und Bäcker, die in der folgenden Reihenfolge ihrer Ränge ankommen: 1, 0, 3, 5, 4, 2. Diese Gala entspricht dem ersten Eingabe- und Ausgabebeispiel.

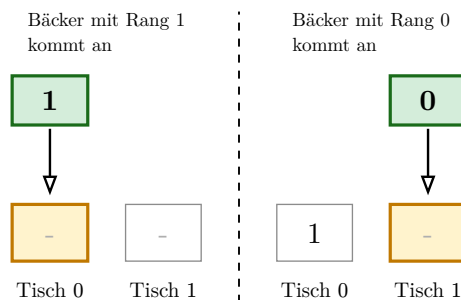


Abbildung 1: Die ersten  $M = 2$  Bäcker legen ihre Pizzen in der Reihenfolge ihrer Ankunft auf die leeren Tische (0, 1).

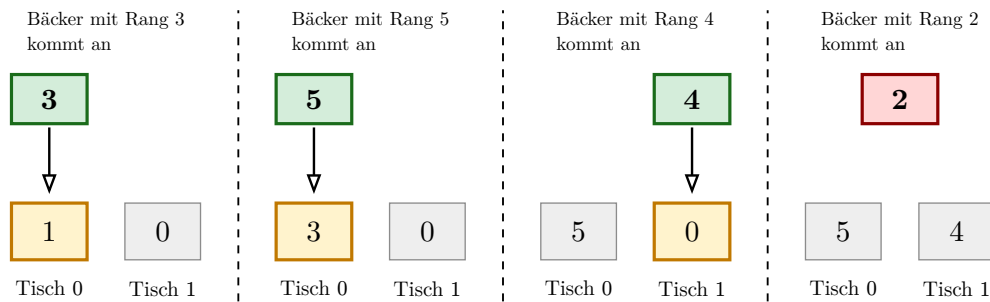


Abbildung 2: Sobald alle Tische belegt sind, geht jeder ankommende Bäcker zum Tisch mit der schlechtesten Pizza, die noch besser ist als seine eigene (durch den Pfeil dargestellt), isst diese Pizza und hinterlässt seine eigene. Wenn keine bessere Pizza existiert, geht der Bäcker frustriert weg (kein Pfeil).

In deinem Artikel möchtest du über die Reihenfolge berichten, in der die Bäcker auf der Pizza-Gala angekommen sind. Leider warst du zu sehr von all den leckeren Pizzen abgelenkt und hast vergessen, die Ankunftsreihenfolge der Bäcker zu notieren. Glücklicherweise kannst du auf jedem Tisch einen Stapel der Tablettis finden, die an diesem Tisch serviert wurden, und zwar in der Reihenfolge, in der die Pizzen serviert wurden.

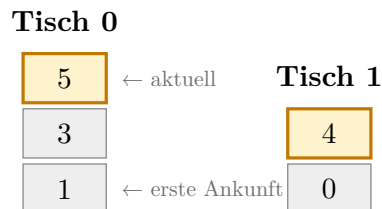


Abbildung 3: Tablettstapel, die dem ersten Beispiel entsprechen. Jeder Stapel listet die Bäcker auf, die in der Reihenfolge ihrer Ankunft an diesem Tisch waren, von unten (erste) bis oben (zuletzt).

Das hervorgehobene Tablett enthält die Pizza, die am Ende der Gala dort zurückgelassen wurde.

Du möchtest diese Informationen verwenden, um die Reihenfolge der ankommenden Bäcker zu rekonstruieren. Du weißt, dass es mehrere mögliche Reihenfolgen gegeben haben könnte. Daher möchtest du für die volle Punktzahl die lexikographisch kleinste gültige Reihenfolge angeben.<sup>1</sup>

## Eingabe

Die erste Zeile enthält zwei Ganzzahlen  $N$  und  $M$ , die Anzahl der Bäcker und die Anzahl der Tische.

Dann folgen  $M$  Zeilen, von denen jede einen Tablettstapel auf einem Tisch beschreibt. Zeile  $i$  beginnt mit einer Ganzzahl  $T_i$ , der Anzahl der Tablettis auf Tisch  $i$ , gefolgt von  $T_i$  Ganzzahlen  $b_{i,j}$ , die den Rang der  $j$ -ten Pizza angeben, die an Tisch  $i$  serviert wurde.

## Ausgabe

Gib NO aus, wenn keine mögliche Reihenfolge die Einschränkungen erfüllt. Gib YES aus, wenn eine mögliche Reihenfolge existiert. Gib in diesem Fall eine zweite Zeile mit  $N$  Ganzzahlen  $a_0, a_1, \dots, a_{N-1}$  aus, den Rängen der Bäcker in der Reihenfolge ihrer Ankunft. Wenn mehrere solcher Permutationen existieren, solltest du die lexikographisch kleinste ausgeben. Beachte, dass teilweise korrekte Antworten möglicherweise Punkte erzielen, wie im Abschnitt Bewertung erläutert.

## Einschränkungen

- $1 \leq M \leq N \leq 300\,000$ .
- $0 \leq b_{i,j} \leq N - 1$ .
- Alle  $b_{i,j}$  sind eindeutig.

<sup>1</sup>Eine Sequenz  $a_0, a_1, \dots, a_{n-1}$  ist lexikographisch kleiner als eine Sequenz  $b_0, b_1, \dots, b_{n-1}$ , wenn es einen Index  $0 \leq t < n$  gibt, sodass  $a_i = b_i$  für alle  $i < t$  und  $a_t < b_t$  gilt.

- $1 \leq T_i \leq N$ .

## Bewertung

Dein Programm wird mit mehreren Testfällen getestet, die in Teilaufgaben gruppiert sind. Um die Punktzahl für eine Teilaufgabe zu erhalten, musst du alle darin enthaltenen Tests korrekt lösen.

⇒ Lösungen, die nur eine korrekte erste Zeile (YES vs NO) liefern, erzielen 20%. Lösungen mit einer korrekten ersten Zeile (YES vs NO) und **irgendeiner gültigen** Reihenfolge (nicht unbedingt der lexikographisch kleinsten), wenn die Antwort YES ist, erzielen weitere 20%. Um die restlichen 60% zu erreichen, musst du die lexikographisch kleinste gültige Reihenfolge ausgeben, wenn die erste Zeile YES ist.

## Teilaufgaben

- **Teilaufgabe 0 [ 0 Punkte]**: Beispiele.
- **Teilaufgabe 1 [20 Punkte]**:  $M = 1$ .
- **Teilaufgabe 2 [10 Punkte]**:  $M = 2$ ,  $N \leq 200$  und die Summe aller  $T_i$  ist  $N$  (mit anderen Worten: kein Bäcker geht frustriert weg).
- **Teilaufgabe 3 [20 Punkte]**:  $M \leq N \leq 200$  und die Summe aller  $T_i$  ist  $N$  (mit anderen Worten: kein Bäcker geht frustriert weg).
- **Teilaufgabe 4 [20 Punkte]**:  $M \leq 10$ .
- **Teilaufgabe 5 [30 Punkte]**: Keine zusätzlichen Einschränkungen.

## Beispiele

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

## Erklärung

Das erste Eingabe- und Ausgabebeispiel entspricht den Abbildungen im Problemtext. Insbesondere ist die Reihenfolge, in der die Bäcker in Abbildung 1 und 2 zur Gala kommen, die lexikographisch kleinste gültige Ankunftsreihenfolge 1, 0, 3, 5, 4, 2.

Im zweiten Beispiel sind die Tablettstapel inkonsistent, da es keine Ankunftsreihenfolge gibt, bei der der Bäcker mit Rang 5 frustriert weggehen würde. Daher ist die Antwort NO.

Im dritten und fünften Beispiel sind die Tablettstapel ebenfalls inkonsistent (keine Ankunftsreihenfolge kann sie erzeugen), daher ist die Antwort NO.

Im vierten Beispiel ( $N = 3$ ,  $M = 1$ ) ist nur eine Ankunftsreihenfolge möglich, nämlich 0, 2, 1.

Beachte im sechsten Beispiel ( $N = 12$ ,  $M = 4$ ), dass die Zahlen 0 und 1 nicht unter den Werten  $b_{i,j}$  vorkommen. Das bedeutet, dass zu einem gewissen Zeitpunkt während der Gala jeder der Bäcker 0 und 1 frustriert weggegangen ist. Die Beispielausgabe zeigt die lexikographisch kleinste gültige Ankunftsreihenfolge. Es existieren weitere gültige Ankunftsreihenfolgen; zum Beispiel 2, 5, 6, 7, 8, 1, 3, 4, 9, 10, 11, 0. Die Ausgabe von YES, gefolgt von einer alternativen gültigen Reihenfolge wie dieser (anstelle der lexikographisch kleinsten), würde als teilweise korrekt für 40% der Punktzahl gewertet werden.