LFSTACK - Lock-Free Stack

题目描述

Sergey 最近正在研究并行数据结构。他尤其喜欢一种叫做并行栈的数据结构。

并行栈也是普通的先入后出栈，但它可以被一个程序的多个线程同时使用，这些线程可以同步地进行压栈、出栈操作。

为了实践他学会的知识，Sergey 实现了这一数据结构。但是他仍然不确定他写出的程序是否可以正确地运作。所以他设计了如下的测试数据：

对每一个线程（编号为 i），有一个含有 A_i 个数字的序列。这一线程将会按顺序把序列中所有的数字都压栈（也就是说，序列的第一个数首先被压栈，之后是第二个，如此继续……）。

Sergey 运行了他的程序，这些数字同步地从不同的线程被压入栈中。当所有的线程都完成工作之后，他希望将栈中所有的数字出栈，并且检查输出的正确性。

但是这并不是一个简单的任务，因为大体上，可能有很多种不同的合法输出，因为不同进程之间压栈操作的顺序并不是确定的。此外，单个进程连续的压栈操作可能被另一进程暂时打断。

例如，即使他只有两个序列中都只有 1 个数字（并且两个进程的数字各不相同）的两个进程，也存在着两种不同的合法输出：有可能是第一个进程先进行压栈；也可能是第二个进程先压栈。

再例如，即使他只有两个进程进行压栈操作，第一个进程有序列(1,2)而第二个进程有序列(3,4)，那么在栈中所有数据出栈之后，输出结果(4,2,3,1)也是合法的，因为：

- 首先，第一个进程将 1 压栈
- 之后，第二个进程将 3 压栈
- 之后，第一个进程将 2 压栈
- 之后，第二个进程将 4 压栈

因为栈是先进后出的数据结构，出栈得到的输出为(4,2,3,1)。

你得到了进程的个数和每个进程的数字序列，还有一个总的输出序列。请判断输出序列是否可能为上述过程进行完毕之后，出栈操作的输出。

输入格式

输入的第一行包含一个整数 T，代表测试数据的组数。

每组测试数据的第一行都包含一个整数 N，代表进程的数量。

之后的 N 行中，每一行都包含了如下的数据：第一个数字为 A_i，代表 i 号进程的数字序列中数字的个数；之后 A_i 个数字 $B_{i,1}, B_{i,2},, B_{i,A_i}$ 代表 i 号进程将会按顺序压栈的数字序列。

每组测试数据的最后一行包含了 $A_1 + A_2 + \cdots + A_N$ 个整数，表示 Sergey 在进行完上述操作之后得到的输出序列。

输出格式

对于每一组测试数据，如果 Sergey 得到的输出序列为合法，则输出一行“`Yes`”；否则输出一行“`No`”。

数据范围和子任务

- $1 \leq T \leq 15$
- $1 \leq A_i$
- $1 \leq B_{i,j} \leq 1000$
- 假设 $P = (A_1 + 1) \times (A_2 + 1) \times \cdot \times (A_N + 1)$，则 $1 \leq P$ 的和 $\leq 10^6$

子任务 1（33 分）：

- $1 \leq P$ 的和 ≤ 1000

子任务 2（11 分）：

- $N = 1$

子任务 3（56 分）：

- 无附加限制

样例数据

输入	输出
2 2 2 1 2 2 3 4 4 3 2 1 2 2 1 2 2 3 4 4 1 2 3	Yes No

样例解释

第一组数据：首先，第一个进程的两个数压栈：1,2。之后，第二个进程的两个数据压栈：3,4。所以入栈顺序为 1,2,3,4；出栈顺序为 4,3,2,1。输出序列合法。

第二组数据：想要得到出栈序列 4,1,2,3，需要入栈序列 3,2,1,4。因此，2 应该在 1 之前入栈。不过，只有第一个进程可以使得这两个数入栈，但 1 一定会在 2 之前入栈，因此这个输出序列一定不合法。

时间限制

1 秒

Problem Setter: Sergey Kulik
Problem Tester: Shah Harshil Ketankumar
Translated by: Hu Zecong