LFSTACK: Lock-Free Stack

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контесте.

Условие:

Сергей недавно изучил потокобезопасные структуры данных. Он очень полюбил структуру данных “потокобезопасный стек”.

Итак, потокобезопасный стек - это обычный стек, который может использоваться несколькими потоками в одной программе. Пусть есть **N** потоков, каждый из которых заносит и извлекает данные из стека одновременно.

Для того, чтобы проверить свои знания, Сергей реализовал эту структуру данных. Но он до сих пор не уверен, работает ли его код правильно. Поэтому он написал тесты следующего типа:

Для каждого потока (скажем, **i**-того) существует массив, состоящий из $\mathbf{A_i}$ целых чисел, которые будут добавлены в стек этим потоком поочередно. То есть, сначала будет добавлено первое число из массива, затем второе и так далее.

Сергей запустил программу и числа были занесены потоками в стек одновременно. Когда все потоки закончили работу, Сережа захотел извлечь все числа из стека и проверить корректность данных.

Но это оказалось непростой задачей, потому что порядок исполнения потоков строго не указан. Более того, один поток может прерывать исполнение, а другой поток продолжить.

Например, если бы у Сергея было только два потока, каждый из которых должен занести по одному числу, то уже существовало бы два возможных варианта корректной последовательности это: числа из первого потока, затем число из второго и наоборот.

Еще один пример: у Сергея два потока, у первого список чисел - **(1, 2)**, у второго - (**3, 4**). В этом случае даже последовательность **(4, 2, 3, 1)** будет корректной т.к:

- Первый поток добавляет первое число **1** в стек
- Второй поток добавляет первое число **3** в стек
- Первый поток добавляет второе число **2** в стек
- Второй поток добавляет второе число **4** в стек

Так как стек - структура данных LIFO (последний зашел, первый вышел), то после извлечения данных получим последовательность **(4, 2, 3, 1)**.

Дано количество потоков, для каждого потока дан массив целых чисел для добавления в стек. Также дана выходная последовательность после извлечения числе из стека. Ваша задача - определить, могла ли эта последовательность быть получена при корректной работе стека.

Формат ввода:

Первая строка содержит единственное целое число **T** – количество тестов.

Далее следует описание тестов в следующем формате:

Первая строка каждого теста содержит единственное целое число **N** – количество потоков.

Каждая из следующих **N** строк содержит описание списка чисел потока в следующем формате: первое целое число **Ai** - количество чисел в списке, затем следуют $\mathbf{A_i}$ разделенных пробелами целых чисел $\mathbf{B_{i,1}}$, $\mathbf{B_{i,2}}$, …, $\mathbf{B_{i,Ai}}$ - числа для добавления в стек **i**-тым потоком.

Последняя строка содержит $\mathbf{A_1+A_2+…+A_N}$ разделенных пробелами целых чисел - извлеченные из стека числа (в порядке извлечения).

Формат вывода:

Для каждого тестового случая выведите в отдельную строку “Yes” (без кавычек), если эта последовательность быть получена при корректной работе стека, иначе выведите “No” (без кавычек).

Ограничения:

- $\mathbf{1 \le T \le 15}$
- $\mathbf{1 \le A_i}$
- $\mathbf{1 \le B_{i,j} \le 1000}$
- Пусть $\mathbf{P = (A_1 + 1) \times (A_2 + 1) \times ... \times (A_N + 1)}$
- **1** ≤ сумма всех $\mathbf{P \le 10^6}$

Подзадачи:

- **Подзадача #1 (33 балла):** **1** ≤ сумма всех $\mathbf{P \le 10^6}$
- **Подзадача #2 (11 баллов): N = 1**
- **Подзадача #3 (56 баллов):** нет дополнительных ограничений

Примеры тестов:

Входные данные:

```
2
2
2 1 2
2 3 4
4 3 2 1
2
2 1 2
2 3 4
4 1 2 3
```

Выходные данные:

```
Yes
No
```

Пояснения:

Тест 1: Сначала первый поток добавил свои числа **1** и **2** в стек, затем второй поток добавил свои числа **3** и **4**. Полученная последовательность **(1, 2, 3, 4)** после извлечения из стека станет последовательностью **(4, 3, 2, 1)**.

Тест 2: Необходимо получить следующую последовательность добавления чисел в стек: **3, 2, 1, 4**. Итак, **2** должно быть добавлено до **1**, но только первый массив содержит эти числа, и в не том порядке, что нам требуется. Отсюда заключаем, что такую последовательность получить невозможно.