



Команды

В классе N школьников, пронумерованных последовательно целыми числами от 0 до $N - 1$. На каждый день у учителя есть задания для школьников этого класса. Каждое задание должно быть выполнено командой школьников в тот же день. Задания могут иметь различную сложность. Для каждого задания учитель знает точное количество человек в команде, которая должна его выполнять.

Разные школьники могут предпочитать различные по количеству человек команды. Школьник с номером i может быть включен только в команду с количеством человек от $A[i]$ до $B[i]$ включительно. Каждый день школьник может быть включен не более, чем в одну команду. Некоторые школьники могут быть не включены ни в одну из команд. Каждая команда выполняет ровно одно задание.

Учитель уже выбрал задания для каждого из следующих Q дней. Для каждого из этих дней определите, возможно ли распределить школьников по командам таким образом, что каждое задание выполняется одной командой.

Пример

Предположим, что в классе 4 школьника и задания выдаются для двух дней. Ограничения на количество человек в команде для школьников задаются в таблице ниже.

Школьник	0	1	2	3
A	1	2	2	2
B	2	3	3	4

В первый день выдается 2 задания. Требуемое количество человек в командах 1 и 3. Эти две команды могут быть сформированы включением школьника с номером 0 в команду из 1 человека, а остальных трех школьников в команду из 3 человек.

Во второй день тоже выдается 2 задания, но в этот раз требуемое количество человек в командах 1 и 1. В этом случае невозможно сформировать команды, так как есть только один школьник, который может быть включен в команду из 1 человека.

Постановка задачи

Вам предоставляется описание всех школьников: N , A и B , а также последовательность из Q запросов, по одному для каждого дня. Каждый запрос состоит из количества заданий для этого дня, обозначенного M , и последовательности K длины M , элементы которой содержат требуемое количество человек в каждой из команд. Для каждого запроса программа должна определить, возможно ли сформировать все команды.

Требуется реализовать функции `init` и `can`:

- `init(N, A, B)` — эта функция будет вызвана первой и ровно один раз.
 - N — количество школьников в классе.
 - A — массив длины N , где $A[i]$ задает минимальное количество человек в команде для школьника с номером i .
 - B — массив длины N , где $B[i]$ задает максимальное количество человек в команде для школьника с номером i .
 - Эта функция не возвращает никакого значения.
 - Во всех подзадачах $1 \leq A[i] \leq B[i] \leq N$ для всех $i = 0, \dots, N - 1$.
- `can(M, K)` — после однократного вызова `init`, эта функция будет вызвана Q раз подряд, по одному разу для каждого дня.
 - M — количество заданий для этого дня.
 - K — массив длины M , элементы которого содержат требуемое количество человек в команде для каждого из этих заданий.
 - Функция должна возвращать 1, если возможно сформировать требуемые команды, и иначе — 0.
 - Во всех подзадачах $1 \leq M \leq N$. Для всех $i = 0, \dots, M - 1$ выполнено $1 \leq K[i] \leq N$. Заметим, что сумма всех $K[i]$ может превосходить N .

Подзадачи

Обозначим сумму всех значений M во всех вызовах функцией `can(M, K)` за S .

Подзадача	Баллы	N	Q	Дополнительные ограничения
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	нет
2	13	$1 \leq N \leq 100\,000$	$Q = 1$	нет
3	43	$1 \leq N \leq 100\,000$	$1 \leq Q \leq 100\,000$	$S \leq 100\,000$
4	23	$1 \leq N \leq 500\,000$	$1 \leq Q \leq 200\,000$	$S \leq 200\,000$

Пример проверяющего модуля

Проверяющий модуль имеет следующий формат входных данных:

- Строка 1: N
- Строки 2, ..., $N + 1$: $A[i] B[i]$
- Строка $N + 2$: Q
- Строки $N + 3, \dots, N + Q + 2$: $M K[0] K[1] \dots K[M - 1]$

Для каждого запроса проверяющий модуль напечатает значения, возвращаемые функцией `cap`.