



Teams

В един клас има N ученици, номерирани с целите числа от 0 до $N - 1$. Всеки ден учителят дава проекти на учениците. Всеки проект трябва да бъде завършен същия ден от отбор от ученици. Проектите са с различна трудност. За всеки проект учителят знае точно колко ученици трябва да има в отбора, който ще работи по проекта.

Всеки ученик предпочита различен брой на участващите в отбора, в който и самият той участва. Т.е. ученик i може да бъде зачислен в отбор, в който броят на участващите е между $A[i]$ и $B[i]$, включително. Във всеки от дните, ученикът може да бъде зачислен към най-много един отбор. Някои ученици може да не са зачислени към отбор. Всеки отбор работи върху един проект.

Учителят е избрал проектите за всеки от следващите Q дни. За всеки от тези дни, вие трябва да определите дали е възможно да бъдат зачислени ученици към отборите така, че да има по един отбор за всеки проект.

Пример

Предполагане, че има $N = 4$ ученици и $Q = 2$ дни. Изискванията на учениците за размерите на отборите са дадени в таблицата:

| ученик | 0 | 1 | 2 | 3 |
|--------|---|---|---|---|
| A | 1 | 2 | 2 | 2 |
| B | 2 | 3 | 3 | 4 |

Нека за първия ден има $M = 2$ проекта и съответните размери на отборите са $K[0] = 1$ и $K[1] = 3$. Те могат да бъдат сформирани по следния начин: Ученик 0 се зачислява към отбора с размер 1 и останалите 3 ученици се зачисляват към отбора с размер 3.

Нека за втория ден има $M = 2$ проекта, но съответните размери на отборите са $K[0] = 1$ и $K[1] = 1$. Сега не е възможно да се сформират отборите, защото само един от учениците иска да участва в отбор с размер 1.

Задача

Дадени са описанията за всички ученици: N , A и B , а също и редицата от Q въпроса — по един за всеки ден. Всеки въпрос се състои от: брой проекти M за деня и редица K с дължина M , съдържаща размерите на отборите. За всеки въпрос вашата програма трябва да съобщи дали е възможно да бъдат сформирани отборите.

Трябва да имплементирате функциите `init` и `can`:

- `init(N, A, B)` — Грейдерът ще извика тази функция само веднъж и ще я извика в

началото.

- N : брой на учениците.
- A : масив с дължина N : $A[i]$ е минималният размер на отбора за ученик i .
- B : масив с дължина N : $B[i]$ е максималният размер на отбора за ученик i .
- Функцията не връща стойност.

Приемате, че $1 \leq A[i] \leq B[i] \leq N$ за всяко $i = 0, \dots, N - 1$.

- $\text{can}(M, K)$ — След извикването на `init`, грейдерът ще извика функцията `can` последователно Q пъти — по веднъж да всеки ден.
 - M : брой на проектите за деня.
 - K : масив с дължина M , съдържащ съответните за проектите размер на отборите.
 - Функцията трябва да върне 1, ако е възможно да се сформират всички отбори; в противен случай функцията трябва да върне 0.

Приемате, че $1 \leq M \leq N$, и че за всяко $i = 0, \dots, M - 1$ имаме $1 \leq K[i] \leq N$.

Отбелязваме, че сумата от всички $K[i]$ N .

Подзадачи

Означаваме с S сумата на стойностите M за всичките извиквания на `can(M, K)`.

| подзадача | точки | N | Q | допълнителни ограничения |
|-----------|-------|-------------------------|-------------------------|--------------------------|
| 1 | 21 | $1 \leq N \leq 100$ | $1 \leq Q \leq 100$ | няма |
| 2 | 13 | $1 \leq N \leq 100,000$ | $Q = 1$ | няма |
| 3 | 43 | $1 \leq N \leq 100,000$ | $1 \leq Q \leq 100,000$ | $S \leq 100,000$ |
| 4 | 23 | $1 \leq N \leq 500,000$ | $1 \leq Q \leq 200,000$ | $S \leq 200,000$ |

Примерен грейдер

Примерният грейдер чете входа в следната последователност

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

За всеки въпрос примерният грейдер отпечатва стойността, която връща `can`.