



Teams

Vi har en klasse med N elever, nummerert 0 til $N - 1$. Hver dag har læreren noen prosjekter til elevene. Hvert prosjekt må utføres av en elevgruppe samme dag. Prosjektene har forskjellig vanskelighetsgrad. Læreren vet nøyaktig hvor stor gruppen som skal jobbe med hvert prosjekt skal være.

Forskjellige elever kan foretrekke forskjellige gruppestørrelser. Elev i kan bare jobbe i grupper med størrelse fra og med $A[i]$ til og med $B[i]$. Hver dag kan hver elev være med i maks én gruppe. Det er mulig at noen av elevene ikke blir satt i noen av gruppene. Hver gruppe vil jobbe med kun ett prosjekt.

Læreren har allerede valgt prosjektene for de neste Q dagene. For hver av disse dagene skal du avgjøre om det er mulig å dele inn elevene i grupper slik at det er en gruppe som jobber med hvert prosjekt.

Eksempel

Anta at det er $N = 4$ elever og $Q = 2$ dager med prosjekter. Elevenes gruppestørrelsepreferanser er gitt i tabellen under.

elev	0	1	2	3
A	1	2	2	2
B	2	3	3	4

Den første dagen er det $M = 2$ prosjekter. Gruppestørrelsene som trengs er $K[0] = 1$ og $K[1] = 3$. Disse to gruppene kan lages ved å ha elev 0 i en gruppe med størrelse 1 og de andre tre elevene i en gruppe med størrelse 3.

Den andre dagen er det igjen $M = 2$ prosjekter, men denne gangen er gruppestørrelsene som trengs $K[0] = 1$ og $K[1] = 1$. I dette tilfellet er det ikke mulig å lage gruppene, siden det er kun én av elevene som kan være med i en gruppe med størrelse 1.

Task

Du får oppgitt beskrivelsen av alle elevene: N , A og B , og en rekke Q spørsmål — et for hver dag. Hvert spørsmål består av antallet M prosjekter og K som er et array med lengde M som inneholder gruppestørrelsene som trengs. For hvert spørsmål må programmet ditt returnere om det er mulig å danne alle gruppene.

Du må implementere funksjonene `init` og `can`:

- `init(N, A, B)` — Graderen vil kalle denne funksjonen først og nøyaktig en gang.
 - N : antall elever.

- A: array med lengde N : $A[i]$ er minste gruppestørrelse for elev i .
 - B: array med lengde N : $B[i]$ er maks gruppestørrelse for elev i .
 - Funksjonen har ingen returverdi
 - Du kan anta at $1 \leq A[i] \leq B[i] \leq N$ for hver $i = 0, \dots, N-1$.
- $\text{can}(M, K)$ — Etter at init er kalt en gang, vil graderen kalle denne funksjonen Q ganger på rad, en gang for hver dag.
- M : antall prosjekter denne dagen.
 - K : et array med lengde M som inneholder gruppestørrelsen som trengs for hvert av prosjektene.
 - Funksjonen skal returnere 1 hvis det er mulig å danne alle gruppene og 0 ellers.
 - Du kan anta at $1 \leq M \leq N$, og at for hver $i = 0, \dots, M-1$ så har vi $1 \leq K[i] \leq N$. Merk at summen av alle $K[i]$ kan være mer enn N .

Subtasks

La S være summen av verdiene i M for alle kallene til $\text{can}(M, K)$.

subtask	points	N	Q	Additional Constraints
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	none
2	13	$1 \leq N \leq 100,000$	$Q = 1$	none
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Sample grader

The sample grader reads the input in the following format:

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

For each question, the sample grader prints the return value of can .