



## Grupper

Det finns en klass med  $N$  elever, numrerade från  $0$  till  $N - 1$ . Varje dag ger klassens lärare ut projekt till eleverna. Alla projekt måste slutföras under dagen av grupper av elever. Projekten kan ha olika svårighetsgrader — för varje projekt vet läraren precis vilken storlek på en grupp som krävs för det.

Olika elever kan föredra olika gruppstorlekar. Mer precist så kan elev  $i$  bara tilldelas till en grupp av storlek mellan  $A[i]$  och  $B[i]$ , inklusive. Under en given dag så kan en elev tilldelas till högst ett projekt. Alla elever behöver inte tilldelas till en grupp. Varje grupp kommer att arbeta på exakt ett projekt.

Läraren har redan bestämt projekten för var och en av de nästföljande  $Q$  dagarna. För alla dessa dagar, bestäm huruvida det är möjligt att tilldela projekten till grupperna på så att alla projekt jobbas på av någon grupp.

## Exempel

Antag att det finns  $N = 4$  elever och  $Q = 2$  dagar. Elevernas krav på gruppstorlekar ges i tabellen nedan.

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

Under första dagen har vi  $M = 2$  projekt. De krävda gruppstorlekarna är  $K[0] = 1$  och  $K[1] = 3$ . Dessa två grupper kan skapas genom att tilldela elev 0 till en grupp av storlek 1, och de resterande eleverna till en grupp av storlek 3.

Under andra dagen har vi återigen  $M = 2$  projekt, men den här gången är storlekskraven  $K[0] = 1$  och  $K[1] = 1$ . I det här fallet är det omöjligt att skapa grupperna, eftersom det bara finns en elev som kan tänka sig att vara i en enpersonsgrupp.

## Uppgift

Du är given beskrivningen av alla eleverna:  $N$ ,  $A$  och  $B$ , och även en sekvens av  $Q$  frågor — en om varje dag. Varje fråga består av ett antal projekt  $M$  för den dagen och en sekvens  $K$  av längd  $M$  som innehåller kraven på gruppstorlekar. För varje fråga måste ditt program returnera huruvida det är möjligt att skapa alla grupperna.

Du skall implementera funktionerna `init` och `can`:

- `init(N, A, B)` — Rättaren kommer att anropa den här funktionen först, och exakt en gång.

- $N$ : antal elever.
  - $A$ : en array av längd  $N$ :  $A[i]$  är den minimala gruppstorleken för elev  $i$ .
  - $B$ : en array av längd  $N$ :  $B[i]$  är den maximala gruppstorleken för elev  $i$ .
  - Funktionen har inget returvärde.
  - Du kan anta att  $1 \leq A[i] \leq B[i] \leq N$  för alla  $i = 0, \dots, N-1$ .
- $\text{can}(M, K)$  — Efter att ha anropat  $\text{init}$  en gång kommer rättaren att anropa den här funktionen  $Q$  gånger i rad, en gång för varje dag.
- $M$ : antal projekt den här dagen.
  - $K$ : en array av längd  $M$  som innehåller kravet på gruppstorlek för var och en av projekten.
  - Funktionen skall returnera 1 om det är möjligt att skapa alla de krävda grupperna, och 0 annars.
  - Du kan anta att  $1 \leq M \leq N$ , och att vi för alla  $i = 0, \dots, M-1$  har  $1 \leq K[i] \leq N$ .  
Notera att summan av alla  $K[i]$  kan överskrida  $N$ .

## Deluppgifter

Låt oss med  $S$  beteckna summan av värden på  $M$  för alla anrop till  $\text{can}(M, K)$ .

deluppgift	poäng	$N$	$Q$	Ytterligare krav
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	inga
2	13	$1 \leq N \leq 100,000$	$Q = 1$	inga
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

## Exemplerrättare

Exemplerrättaren läser indata på följande format:

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

För varje fråga skriver exemplerrättaren ut returvärdet på  $\text{can}$ .