



Team

In una classe sono presenti N studenti, numerati da 0 a $N - 1$. Ogni giorno, l'insegnante porta dei progetti da far svolgere ai suoi studenti. Ciascun progetto deve essere completato in giornata da un team di studenti, e l'insegnante conosce per ogni progetto qual è la grandezza perfetta del team che dovrà lavorarci.

Ovviamente, gli studenti hanno delle preferenze sulla grandezza del team in cui capiteranno. In particolare, lo studente i accetterà di essere assegnato ad un team a condizione che la sua dimensione sia compresa tra $A[i]$ e $B[i]$ (estremi inclusi). In una stessa giornata, uno studente può essere assegnato ad un unico team. Alcuni studenti possono anche rimanere non assegnati ad alcun team, e un team lavora sempre ad un unico progetto.

L'insegnante ha già pianificato i progetti da svolgere nei prossimi Q giorni. Aiuta l'insegnante a determinare, per ciascun giorno, se è possibile o meno assegnare gli studenti a dei team in modo tale da avere sempre un team al lavoro su ciascun progetto.

Esempio

Supponiamo ci siano $N = 4$ studenti e $Q = 2$ giorni. Le preferenze degli studenti sulle dimensioni dei team sono riportate nella seguente tabella:

studente	0	1	2	3
A	1	2	2	2
B	2	3	3	4

Nel primo giorno vanno svolti $M = 2$ progetti. Le dimensioni dei team richieste dall'insegnante sono $K[0] = 1$ e $K[1] = 3$. Questi due team possono essere formati assegnando lo studente 0 al team di dimensione 1 e i tre studenti rimanenti al team di dimensione 3 .

Nel secondo giorno ci sono di nuovo $M = 2$ progetti, ma questa volta le dimensioni richieste sono $K[0] = 1$ e $K[1] = 1$. Questa volta non è possibile formare i team, dato che c'è un solo studente disposto a stare in un team di dimensione 1 .

Implementazione

Ti viene fornita una descrizione di tutti gli studenti: N , A , e B ; assieme alla sequenza delle Q query — una per giorno. Ciascuna query consiste di un numero M di progetti da realizzare nella giornata e di una sequenza K di lunghezza M contenente le dimensioni richieste per i team. Per ogni query, dovrai calcolare se è possibile o meno formare tutti i gruppi.

Dovrai implementare le funzioni `init` e `can`:

- `init(N, A, B)` — il grader chiamerà questa funzione esattamente una volta.
 - N : il numero di studenti.
 - A : un array di lunghezza N , in cui $A[i]$ è la minima dimensione di un team accettata dallo studente i .
 - B : un array di lunghezza N , in cui $B[i]$ è la massima dimensione di un team accettata dallo studente i .
 - Questa funzione non ha valore di ritorno.

Puoi assumere che $1 \leq A[i] \leq B[i] \leq N$ per ogni $i = 0, \dots, N - 1$.

- `can(M, K)` — dopo la chiamata a `init`, il grader chiamerà questa funzione Q volte di seguito, una per giorno.
 - M : il numero di progetti in questo giorno.
 - K : un array di lunghezza M contenente le dimensioni richieste per i team associati a ciascun progetto.
 - La funzione dovrebbe restituire 1 se è possibile formare tutti i team, 0 altrimenti.
 - Puoi assumere che $1 \leq M \leq N$, e che $1 \leq K[i] \leq N$ per ogni $i = 0, \dots, M - 1$. Invece, la somma di tutti i $K[i]$ potrebbe superare N .

Subtask

Sia S la somma dei valori di M in tutte le chiamate a `can(M, K)`.

subtask	punti	N	Q	Restrizioni aggiuntive
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	nessuna
2	13	$1 \leq N \leq 100\,000$	$Q = 1$	nessuna
3	43	$1 \leq N \leq 100\,000$	$1 \leq Q \leq 100\,000$	$S \leq 100\,000$
4	23	$1 \leq N \leq 500\,000$	$1 \leq Q \leq 200\,000$	$S \leq 200\,000$

Grader di prova

Il grader di prova legge l'input nel seguente formato:

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

Per ciascuna query, il grader di prova stampa in output il valore di ritorno di `can`.