



Tímy

V triede je N študentov, ktorí sú očíslovaní od 0 po $N - 1$. Učiteľ si na každý deň pripraví nejaké projekty pre študentov. Každý projekt musí byť kompletne vykonaný tímom študentov v ten istý deň, v ktorý bol zadaný. Projekty sú rôzne ťažké, a preto učiteľ pre každý projekt určil veľkosť tímu, ktorý na ňom má pracovať.

Každý študent preferuje rôzne veľké tímy. Presnejšie, i -tý študent môže byť zaradený iba do tímov veľkosti od $A[i]$ po $B[i]$ vrátane. Študent môže byť každý deň zaradený najviac do jedného tímu. Niektorí študenti nemusia byť zaradení do tímu vôbec. Každý tím môže v jeden deň pracovať len na jednom projekte.

Učiteľ už pripravil projekty na každý z nasledujúcich Q dní. Pre každý z týchto dní je potrebné určiť, či je možné zaradiť študentov do tímov tak, aby na každom projekte pracoval nejaký tím.

Príklad

Predpokladajme, že sú $N = 4$ študenti a $Q = 2$ dni. Ohraničenia pre veľkosť tímu každého študenta sú dané v nižšie uvedenej tabuľke.

študent	0	1	2	3
A	1	2	2	2
B	2	3	3	4

Na prvý deň sú pripravené $M = 2$ projekty. Požadovaná veľkosť tímov je $K[0] = 1$ a $K[1] = 3$. Potrebné dva tímy môžu byť vytvorené zaradením študenta 0 do tímu veľkosti 1 a zostávajúcich troch študentov do tímu veľkosti 3 .

Na druhý deň sú pripravené opäť $M = 2$ projekty, ale požadované veľkosti tímov sú $K[0] = 1$ a $K[1] = 1$. V tomto prípade nie je možné vytvoriť vhodné tímy, pretože existuje len jeden študent, ktorý je ochotný pracovať v tíme veľkosti 1 .

Úloha

Sú dané požiadavky všetkých študentov: N , A , a B , a tiež postupnosť Q otázok — jedna otázka na každý deň. Každá otázka pozostáva z M projektov určených na daný deň a postupnosti K dĺžky M , ktorá obsahuje požadované veľkosti tímov. Váš program musí dať odpoveď na každú otázku, t. j. pre každý deň povedať, či je možné vytvoriť všetky požadované tímy.

Implementujte funkcie `init()` a `can()`:

- `init(N, A, B)` — Grader volá túto funkciu ako prvú a volá ju práve raz.
- N : počet študentov.

- A: pole dĺžky N : $A[i]$ je minimálna veľkosť tímu, v ktorom môže študent i pracovať.
- B: pole dĺžky N : $B[i]$ je maximálna veľkosť tímu, v ktorom môže študent i pracovať.
- Funkcia nemá žiadnu návratovú hodnotu.

Predpokladajte, že platí $1 \leq A[i] \leq B[i] \leq N$ pre každé $i = 0, \dots, N-1$.

- $\text{can}(M, K)$ — Po jedinom volaní $\text{init}()$, grader volá túto funkciu Q -krát za sebou, jedenkrát pre každý deň.
 - M : počet projektov pre daný deň.
 - K : pole dĺžky M , ktoré obsahuje požadované veľkosti tímov pre projekty.
 - Funkcia vráti 1 , ak je možné vytvoriť všetky požadované tímy, inak vráti hodnotu 0 .
 - Predpokladajte, že $1 \leq M \leq N$, a pre každé $i = 0, \dots, M-1$ máme $1 \leq K[i] \leq N$.
Poznamenávame, že súčet všetkých $K[i]$ môže byť väčší ako N .

Podúloha

Nech S je súčet hodnôt M vo volaniach $\text{can}(M, K)$.

podúloha	body	N	Q	d'alšie ohraničenia
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	žiadne
2	13	$1 \leq N \leq 100\,000$	$Q = 1$	žiadne
3	43	$1 \leq N \leq 100\,000$	$1 \leq Q \leq 100,000$	$S \leq 100\,000$
4	23	$1 \leq N \leq 500\,000$	$1 \leq Q \leq 200,000$	$S \leq 200\,000$

Ukážkový grader

Grader číta vstup v nasledujúcom tvare:

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

Pre každú otázku grader vypíše návratovú hodnotu funkcie $\text{can}()$.