

تیم‌ها

کلاسی را در نظر بگیرید شامل N دانش‌آموز که با اعداد 0 تا $N - 1$ شماره‌گذاری شده‌اند. معلم کلاس هر روز تعدادی پروژه به این دانش‌آموزان می‌دهد. هر کدام از این پروژه‌ها باید در همان روز توسط یک تیم از دانش‌آموزان انجام شود. میزان سختی پروژه‌ها متفاوت است و معلم می‌داند که تیم تخصیص داده شده برای هر کدام از پروژه‌ها باید دقیقاً شامل چند نفر باشد.

دانش‌آموزان مختلف ممکن است ترجیح دهند در تیم‌های با اندازه‌های متفاوتی باشند. به بیان دقیق‌تر، دانش‌آموز i باید به تیمی با اندازه‌ی حداقل $A[i]$ و حداکثر $B[i]$ تخصیص داده شود. هر دانش‌آموز هر روز می‌تواند حداکثر به یک تیم تخصیص داده شود. بعضی از دانش‌آموزان ممکن است به هیچ تیمی تخصیص داده نشوند. هر تیم روی دقیقاً یک پروژه کار می‌کند.

معلم برای هر یک از Q روز آینده تعدادی پروژه انتخاب کرده است. برای هر یک از این روزها، تعیین کنید که آیا می‌توان دانش‌آموزان را به پروژه‌های مربوط به آن روز به گونه‌ای تخصیص داد که هر پروژه دقیقاً یک تیم داشته باشد.

مثال

فرض کنید $N = 4$ دانش‌آموز و $Q = 2$ روز داریم. محدودیت اندازه‌ی تیم‌ها برای هر کدام از این دانش‌آموزان در جدول زیر داده شده است:

دانش‌آموز	0	1	2	3
A	1	2	2	2
B	2	3	3	4

در روز اول $M = 2$ پروژه وجود دارند و اندازه‌ی تیم لازم برای این پروژه‌ها $K[0] = 1$ و $K[1] = 3$ است. یک راه‌حل ممکن برای این روز این است که دانش‌آموز 0 به تیمی با اندازه‌ی 1 و باقی دانش‌آموزان به تیمی با اندازه‌ی 3 تخصیص داده شوند.

در روز دوم هم $M = 2$ پروژه وجود دارند، ولی این بار اندازه‌ی تیم لازم برای این پروژه‌ها $K[0] = 1$ و $K[1] = 1$ است. چون فقط یکی از دانش‌آموزان می‌تواند در تیمی با اندازه‌ی 1 باشد، مسئله برای این روز جوابی ندارد.

مسئله

اطلاعات مربوط به دانش‌آموزان (N ، A و B) و همچنین Q سؤال (یکی برای هر روز) به شما داده شده است. هر سؤال شامل تعداد پروژه‌های آن روز (M) و همچنین دنباله‌ی K متشکل از M عدد است که اندازه‌ی تیم‌ها را مشخص می‌کنند. برای هر سؤال، برنامه‌ی شما باید تعیین کند که آیا شکل‌دهی این تیم‌ها امکان‌پذیر است یا خیر.

شما باید دو تابع `init` و `can` را پیاده‌سازی کنید:

• $\text{init}(N, A, B) -$ این تابع، اول کار دقیقاً یک بار توسط ارزیاب فراخوانی می‌شود.

• N : تعداد دانش‌آموزان

• A : آرایه‌ای به طول N : $A[i]$ حداقل اندازه‌ی تیم برای دانش‌آموز i است.

• B : آرایه‌ای به طول N : $B[i]$ حداکثر اندازه‌ی تیم برای دانش‌آموز i است.

• این تابع چیزی بر نمی‌گرداند.

• می‌توانید فرض کنید که برای هر i از 0 تا $N - 1$ داریم $1 \leq A[i] \leq B[i] \leq N$.

• $\text{can}(M, K) -$ پس از این که init یک بار فراخوانی شد، ارزیاب این تابع را Q بار پشت سر هم (یک بار برای هر روز) فراخوانی می‌کند.

• M : تعداد پروژه‌های این روز

• K : آرایه‌ای به طول M شامل اندازه‌ی تیم‌های مورد نیاز برای هر کدام از این پروژه‌ها.

• این تابع باید در صورتی که شکل‌دهی همه‌ی این تیم‌ها ممکن باشد 1 و در غیر این صورت 0 برگرداند.

• می‌توانید فرض کنید که $1 \leq M \leq N$ و برای هر $i = 0, \dots, M - 1$ داریم $1 \leq K[i] \leq N$. توجه کنید که جمع همه‌ی $K[i]$ ‌ها ممکن است بیش‌تر از N باشد.

زیرمسئله‌ها

جمع همه‌ی مقادیر M در تمام دفعاتی که تابع $\text{can}(M, K)$ فراخوانی می‌شود را با S نشان می‌دهیم.

زیرمسئله	امتیاز	N	Q	محدودیت‌های دیگر
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	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

• خط ۱: N

• خطوط ۲ تا $N+1$: $A[i]$ و سپس $B[i]$

• خط $N+2$: Q

• خطوط $N+3$ تا $N+Q+2$: مقادیر M و سپس $K[0]$ تا $K[M-1]$

برای هر سؤال، ارزیاب نمونه مقدار خروجی تابع can را چاپ می‌کند.