



International Olympiad in Informatics 2015

26th July - 2nd August 2015

Almaty, Kazakhstan

Day 1

teams

Language: tr-TR

Gruplar

Bir sınıfta 0 ile $N - 1$ arasında numaralanmış toplam N tane öğrenci vardır. Öğretmen her gün öğrencilere projeler vermektedir. Her bir proje, verildiği gün içinde bir öğrenci grubu ile tamamlanmalıdır. Projelerin farklı zorluk seviyeleri vardır. Öğretmen, her bir proje üzerinde çalışacak gruptaki öğrenci sayısını tam olarak bilmektedir.

Farklı öğrenciler, değişik grup eleman sayıları ile çalışmak istemektedir. Daha açık olarak, i öğrencisi, sadece eleman sayısı $A[i]$ ve $B[i]$ arasında olan bir gruba atanabilir (sınırlar dahildir). Her gün bir öğrenci en fazla bir gruba atanabilir. Bazı öğrenciler hiç bir gruba atanmayabilir. Her bir grup tek bir proje üzerinde çalışacaktır.

Gelecek Q günün her biri için, öğretmen projeleri seçmiştir. Bu günlerin her birinde, her bir proje üzerinde bir grubun çalışmasını sağlamak için öğrencileri gruplara atamanın mümkün olup olmadığını tespit edin.

Örnek

$N = 4$ tane öğrenci ve $Q = 2$ gün olsun. Öğrencilerin grup eleman sayıları ile ilgili kısıtları aşağıdaki tabloda verilmiştir.

öğrenci	0	1	2	3
A	1	2	2	2
B	2	3	3	4

İlk günde $M = 2$ tane proje vardır. Gereken grup eleman sayıları $K[0] = 1$ ve $K[1] = 3$. Bu iki grup, 0 öğrencisini eleman sayısı 1 olan bir gruba ve geri kalan üç öğrenciyi eleman sayısı 3 olan gruba atayarak elde edilebilir.

İkinci günde yine $M = 2$ tane proje vardır, fakat bu sefer gereken takım eleman sayıları $K[0] = 1$ ve $K[1] = 1$ olarak karşımıza çıkmaktadır. Bu durumda, grupları oluşturmak mümkün değildir çünkü eleman sayısı 1 olan grupta olabilecek sadece tek bir öğrenci vardır.

Görev

Tüm öğrencilerin tanımları verilmektedir: N , A , ve B , ile aynı zamanda her gün için bir tane olmak üzere Q tane soru dizisi. Her bir soru, o gün için olan proje sayısı M ve M uzunluğunda gereken grup eleman sayısını içeren K dizisi. Her bir soru için, programınız tüm grupları oluşturmanın mümkün olup olmadığını dönmelidir.

`init` ve `can` fonksiyonlarınızı gerçekleştirmeniz gerekmektedir:

- `init(N, A, B)` — Grader bu fonksiyonu başlangıçta ve sadece bir defa çağıracaktır.

- N : öğrenci sayısı.
- A : N uzunluğunda bir dizi: $A[i]$, i öğrencisinin atanabileceği minimum grup eleman sayısıdır.
- B : N uzunluğunda bir dizi: $B[i]$, i öğrencisinin atanabileceği maksimum grup eleman sayısıdır.
- Fonksiyon bir değer geri dönmaz.

Her bir $i = 0, \dots, N-1$ için $1 \leq A[i] \leq B[i] \leq N$ olduğunu kabul edebilirsiniz.

- $can(M, K)$ — $init$ fonksiyonunu bir kez çağırdıktan sonra, grader bu fonksiyonu her biri bir gün için olmak üzere ard arda Q defa çağıracaktır.
 - M : Bu gün için proje sayısı.
 - K : Bu projelerin her biri için gereken grup eleman sayısını içeren M uzunluğundaki dizi.
 - Eğer tüm gereken grupları oluşturmak mümkünse fonksiyon 1, değilse 0 dönmelidir.
 - $1 \leq M \leq N$ olduğunu, ve her bir $i = 0, \dots, M-1$ için, $1 \leq K[i] \leq N$ olduğunu kabul edebilirsiniz. Tüm $K[i]$ toplamının N 'yi geçebileceğine dikkat edin.

Alt görevler

$can(M, K)$ 'ya olan tüm çağrılar içindeki M değerlerinin toplamını S ile gösterelim.

alt görev	puan	N	Q	Ek Kısıtlar
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	yok
2	13	$1 \leq N \leq 100,000$	$Q = 1$	yok
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Örnek grader

Örnek grader girdiyi aşağıdaki formatta okur:

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

Her bir soru için, örnek grader can 'm geri gönen değerini yazmalıdır.