



International Olympiad in Informatics 2014

13-20th July 2014

Taipei, Taiwan

Day-2 tasks

holiday

Language: tr-TR

Tatil

Jian-Jia bir sonraki tatilini Tayvan'da geçirmek için plan yapmaktadır. Tatili süresince, Jian-Jia şehirler arasında gezerek şehirlerdeki eğlence yerlerini görmektedir.

Tayvan'da tek bir yol üzerinde bulunan toplam n tane şehir vardır. Şehirler ardışık olarak 0 ile $n - 1$ arasında numaralandırılmıştır. Şehir i için, $0 < i < n - 1$ olmak üzere, komşu şehirler $i - 1$ ve $i + 1$ 'dir. Şehir 0'a komşu olan tek şehir, Şehir 1'dir. Şehir $n - 1$ 'e komşu olan tek şehir de, Şehir $n - 2$ 'dir.

Her bir şehirde belirli sayıda eğlence yerleri bulunmaktadır. Jian-Jia'nın tatili d günlüktür ve o tatilinde mümkün olan en fazla sayıda eğlence yeri görmek istemektedir. Jian-Jia tatiline başlayacağı şehri önceden seçmiştir. Tatilin her bir gününde, Jian-Jia, ya komşu bir şehre geçmektedir, ya da bulunduğu şehrin tüm eğlence yerlerini görmektedir, fakat her ikisini birden değil. Jian-Jia aynı şehirden birden fazla geçse bile, *o şehrin eğlence yerlerini hiç bir zaman iki kez görmeyecektir*. Jian-Jia'nın tatilinde mümkün olduğunca çok sayıda farklı eğlence yerlerini görmesi için lütfen yardım ediniz.

Örnek

Jian-Jia'nın tatili 7 günlük olsun ve aşağıdaki tabloda verildiği gibi 5 tane şehir bulunsun. Tatile şehir 2'den başlasın. İlk günde, Jian-Jia, şehir 2'deki 20 eğlence yerini görür. İkinci günde, şehir 2'den şehir 3'e geçer ve üçüncü günde şehir 3'deki 30 eğlence yerini görür. Sonraki 3 günde, Jian-Jia şehir 3'den şehir 0'a gider ve yedinci günde şehir 0'daki 10 eğlence yerini görür. Bu şekilde Jian-Jia'nın gördüğü eğlence yerleri sayısı $20 + 30 + 10 = 60$ olur ki bu sayı Jian-Jia'nın tatiline şehir 2'den başlaması halinde 7 günde görebileceği maksimum eğlence yeri sayısıdır.

şehir	eğlence yeri sayısı
0	10
1	2
2	20
3	30
4	1

gün	faaliyet
1	şehir 2'deki eğlence yerlerini gör
2	şehir 2'den şehir 3'e git
3	şehir 3'deki eğlence yerlerini gör
4	şehir 3'den şehir 2'ye git
5	şehir 2'den şehir 1'e git
6	şehir 1'den şehir 0'a git
7	şehir 0'daki eğlence yerlerini gör

Görev

Jian-Jia'nın görebileceği maksimum eğlence yeri sayısını hesaplayan `findMaxAttraction` fonksiyonunu yazınız.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: şehir sayısı.
 - `start`: başlangıç şehrinin numarası.
 - `d`: Gün sayısı
 - `attraction`: n uzunluğunda dizi; `attraction[i]` şehir i 'deki eğlence yeri sayısıdır, $0 \leq i \leq n - 1$ olmak üzere.
 - Fonksiyon Jian-Jia'nın görebileceği maksimum eğlence yeri sayısını döndürmelidir.

Altgörevler

Tüm altgörevlerde $0 \leq d \leq 2n + \lfloor n/2 \rfloor$ ve her bir şehirdeki eğlence yeri sayısı negatif değildir.

Ek kısıtlar:

altgörev	puan	n	şehirdeki maksimum eğlence yeri sayısı	başlangıç şehri
1	7	$2 \leq n \leq 20$	1,000,000,000	kısıt yok
2	23	$2 \leq n \leq 100,000$	100	şehir 0
3	17	$2 \leq n \leq 3,000$	1,000,000,000	kısıt yok
4	53	$2 \leq n \leq 100,000$	1,000,000,000	kısıt yok

Gerçekleştirim detayları

İsmi `holiday.c`, `holiday.cpp` veya `holiday.pas` olan tek bir dosya göndermelisiniz. Bu dosya yukarıda tanımlanan ve aşağıda formatı verilen altprogramı içermelidir. C/C++ programı için `holiday.h` header dosyasını da eklemelisiniz.

Sonucun büyük olabileceğine dikkat edin, ve `findMaxAttraction` fonksiyonunun dönen değerinin tipi 64-bit tamsayıdır.

C/C++ program

```
long long int findMaxAttraction(int n, int start, int d,
int attraction[]);
```

Pascal program

```
function findMaxAttraction(n, start, d : longint;
attraction : array of longint): int64;
```

Örnek grader

Örnek grader girdiyi aşağıdaki formatta okur:

- satır 1: n , $start$, d .
- satır 2: $attraction[0], \dots, attraction[n-1]$.

Örnek değerlendirici `findMaxAttraction` 'ın dönen değerini yazacaktır.