



Kayık

Mao-Kong Kayıkları Taipei'de ünlü bir eğlence yeridir. Kayık sistemi, dairesel bir raydan ve bu ray üzerinde sabit bir yönde 1'den n 'e kadar numaralandırılmış n tane kayıktan oluşmaktadır. i . kayık istasyonu geçtikten sonra eğer $i < n$ ise istasyonu geçen bir sonraki kayık $i + 1$. kayık olur ya da eğer $i = n$ ise 1. kayık olur.

Kayıklar bazen bozulabilir. Neyse ki elimizde $n + 1$, $n + 2$, ve devamı şeklinde numaralandırılmış sonsuz sayıda yedek kayık bulunmaktadır. Bir kayık bozulduğunda, bozuk kayık çıkarılarak bozuk kayığın rayda bulunduğu yere sıradaki ilk uygun yedek kayık (yani en küçük numaralı kullanılmamış yedek kayık) yerleştirilir. Örnek olarak, eğer beş kayık varsa ve 1. kayık bozulursa, 1. kayığı 6. kayık ile değiştiririz.

İstasyonda oturup, geçen kayıkları seyretmek sevdiğiniz aktivitelerden birisidir. Bir *kayık dizisi* istasyondan geçen n adet kayığın numaralarının dizisidir. Siz kayıkları izlemeye başlamadan önce bazı kayıklar bozulmuş ve değiştirilmiş olabilir, ama siz izlemeye başladıktan sonra hiçbir kayık bozulmaz.

Bu arada, aynı kayık konfigürasyonunun, siz izlemeye başladığınızda istasyondan ilk geçen kayığa bağlı olarak, birden fazla kayık dizisi oluşturabileceğine dikkat edin. Örnek olarak, eğer hiçbir kayık bozulmamışsa hem (2, 3, 4, 5, 1) hem de (4, 5, 1, 2, 3) olası kayık dizileri olabilirler, fakat (4, 3, 2, 5, 1) olası bir kayık dizisi değildir (çünkü bu dizide kayıklar yanlış sıradadır).

Eğer 1. kayık bozulursa, (4, 5, 6, 2, 3) kayık dizisini gözlemleyebiliriz. Sonra 4. kayık bozulursa, onu 7. kayık ile değiştiririz ve (6, 2, 3, 7, 5) kayık dizisini gözlemleyebiliriz. Hemen sonrasında 7. kayık bozulursa, onu 8. kayık ile değiştiririz ve (3, 8, 5, 6, 2) kayık dizisini gözlemleyebiliriz.

| bozuk kayık | yeni kayık | olası kayık dizisi |
|-------------|------------|--------------------|
| 1 | 6 | (4, 5, 6, 2, 3) |
| 4 | 7 | (6, 2, 3, 7, 5) |
| 7 | 8 | (3, 8, 5, 6, 2) |

Bir *bozulma dizisi*, bozuldukları sıra ile, bozulan kayıkların numaralarının serisidir. Bir önceki örnekte bozulma dizisi (1, 4, 7)'dir. Bir r bozulma dizisi, eğer kayıklar r 'de belirtilen sırada bozulduğunda g gözlemlenebiliyorsa, g kayık dizisini *üretir*.

Kayık Dizisi Kontrolü

İlk üç altgörevde, bir girdi dizisinin bir kayık dizisi olup olamayacağını kontrol edeceksiniz. Aşağıdaki tabloda kayık dizisi olan ve olmayan örnek diziler görebilirsiniz. `valid` adında bir fonksiyon gerçekleştirmelisiniz.

- `valid(n, inputSeq)`

- n : girdi dizisinin boyu.
- $inputSeq$: n uzunluğunda bir dizi; $inputSeq[i]$ girdi dizisinin i . elemanıdır, $0 \leq i \leq n - 1$.
- Girdi dizisi bir kayak dizisi ise fonksiyon 1, değil ise 0 dönmelidir.

1, 2, 3 no'lu Altgörevler

| altgörev | puanı | n | $inputSeq$ |
|----------|-------|------------------|---|
| 1 | 5 | $n \leq 100$ | 1'den n 'ye kadar bütün değerler tam olarak bir kere dizide geçer |
| 2 | 5 | $n \leq 100,000$ | $1 \leq inputSeq[i] \leq n$ |
| 3 | 10 | $n \leq 100,000$ | $1 \leq inputSeq[i] \leq 250,000$ |

Örnekler

| altgörev | $inputSeq$ | dönen değer | not |
|----------|-----------------------|-------------|-------------------------------|
| 1 | (1, 2, 3, 4, 5, 6, 7) | 1 | |
| 1 | (3, 4, 5, 6, 1, 2) | 1 | |
| 1 | (1, 5, 3, 4, 2, 7, 6) | 0 | 1, 5'ten hemen önce gelemez |
| 1 | (4, 3, 2, 1) | 0 | 4, 3'ten hemen önce gelemez |
| 2 | (1, 2, 3, 4, 5, 6, 5) | 0 | numarası 5 olan iki kayak var |
| 3 | (2, 3, 4, 9, 6, 7, 1) | 1 | bozulma dizisi (5, 8) |
| 3 | (10, 4, 3, 11, 12) | 0 | 4, 3'ten hemen önce gelemez |

Bozulma Dizisi

Bir sonraki üç altgörevde, verilen bir kayak dizisini üretebilecek olası bir bozulma dizisi oluşturmalısınız. Verilen kayak dizisini üretebilen herhangi bir bozulma dizisi çözüm olarak kabul edilecektir. `replacement` adında bir fonksiyon gerçekleştirmelisiniz.

- `replacement(n, gondolaSeq, replacementSeq)`
 - n kayak dizisinin boyudur.
 - $gondolaSeq$: n uzunluğunda dizi; $gondolaSeq$ 'in doğru bir kayak dizisi olduğu garantidir, ve $gondolaSeq[i]$ dizinin i . elemanıdır, $0 \leq i \leq n - 1$.
 - Fonksiyon, bozulma dizisinin uzunluğunu, l 'yi, dönmelidir.
 - $replacementSeq$: bozulma dizisinin tamamını yerleştirebilecek büyüklükte bir dizi; bulduğunuz bozulma dizisinin i . elemanı $replacementSeq[i]$ 'de olacak şekilde yerleştirmelisiniz, $0 \leq i \leq l - 1$.

4, 5, 6 no'lu Altgörevler

| altgörev | puan | n | gondolaSeq |
|----------|------|------------------|--|
| 4 | 5 | $n \leq 100$ | $1 \leq \text{gondolaSeq}[i] \leq n + 1$ |
| 5 | 10 | $n \leq 1,000$ | $1 \leq \text{gondolaSeq}[i] \leq 5,000$ |
| 6 | 20 | $n \leq 100,000$ | $1 \leq \text{gondolaSeq}[i] \leq 250,000$ |

Örnekler

| altgörev | gondolaSeq | dönen değer | replacementSeq |
|----------|-----------------------|-------------|----------------|
| 4 | (3, 1, 4) | 1 | (2) |
| 4 | (5, 1, 2, 3, 4) | 0 | () |
| 5 | (2, 3, 4, 9, 6, 7, 1) | 2 | (5, 8) |

Bozulma Dizilerini Sayma

Bir sonraki dört altgörevde, verilen bir diziyi (doğru bir kayak dizisi olabilir ya da olmayabilir) üretebilen bütün olası bozulma dizilerinin sayısını mod **1,000,000,009**'da bulmalısınız. `countReplacement` adında bir fonksiyon gerçekleştirmelisiniz.

- `countReplacement(n, inputSeq)`
 - n : girdi dizisinin boyu.
 - `inputSeq`: n uzunluğunda bir dizi; `inputSeq[i]` girdi dizisinin i . elemanıdır, $0 \leq i \leq n - 1$.
 - Eğer girdi dizisi bir kayak dizisi ise, bu kayak dizisini üretebilecek bütün bozulma dizilerinin sayısını bulmalısınız (bu sayı çok çok büyük olabilir), ve bu sayıyı mod **1,000,000,009**'da dönmelisiniz. Eğer girdi dizisi bir kayak dizisi değil ise, fonksiyonunuz 0 dönmelidir. Eğer girdi dizisi bir kayak dizisi ise fakat hiçbir kayak bozulmamış ise, o zaman fonksiyon 1 dönmelidir.

7, 8, 9, 10 no'lu Altgörevler

| altgörev | puan | n | inputSeq |
|----------|------|--------------------|---|
| 7 | 5 | $4 \leq n \leq 50$ | $1 \leq \text{inputSeq}[i] \leq n + 3$ |
| 8 | 15 | $4 \leq n \leq 50$ | $1 \leq \text{inputSeq}[i] \leq 100$, ve $1, \dots, n$ no'lu ilk kayıklardan en az $n - 3$ tanesi bozulmamıştır. |
| 9 | 15 | $n \leq 100,000$ | $1 \leq \text{inputSeq}[i] \leq 250,000$ |
| 10 | 10 | $n \leq 100,000$ | $1 \leq \text{inputSeq}[i] \leq 1,000,000,000$ |

Örnekler

| altgörev | inputSeq | dönen değer | bozulma dizisi |
|----------|------------------------|-------------|------------------------------------|
| 7 | (1, 2, 7, 6) | 2 | (3, 4, 5) or (4, 5, 3) |
| 8 | (2, 3, 4, 12, 6, 7, 1) | 1 | (5, 8, 9, 10, 11) |
| 9 | (4, 7, 4, 7) | 0 | inputSeq bir kayak dizisi değildir |
| 10 | (3, 4) | 2 | (1, 2) or (2, 1) |

Gerçekleştirim detayları

İsmi `gondola.c`, `gondola.cpp` ya da `gondola.pas` olan tek bir dosya göndermelisiniz. Bu dosya, yukarıda tanımlanan üç fonksiyonun hepsini (altgörevlerden sadece bir kısmını yapmış olsanız bile) aşağıdaki formatta gerçekleştirmelidir. C/C++ gerçekleştirmeleri `gondola.h` header dosyasını include etmelidir.

C/C++ programları

```
int valid(int n, int inputSeq[]);
int replacement(int n, int gondolaSeq[], int replacementSeq[]);
int countReplacement(int n, int inputSeq[]);
```

Pascal programları

```
function valid(n: longint; inputSeq: array of longint): integer;
function replacement(n: longint; gondolaSeq: array of longint;
var replacementSeq: array of longint): longint;
function countReplacement(n: longint; inputSeq: array of longint):
longint;
```

Örnek grader

Örnek grader girdiyi aşağıdaki formatta okur:

- satır 1: T , programınızın çözmesi beklenen altgörev numarası ($1 \leq T \leq 10$).
- satır 2: n , girdi dizisinin boyu.
- satır 3: Eğer T 4, 5, ya da 6 ise, bu satır `gondolaSeq[0], ..., gondolaSeq[n-1]`'i içerir. Değil ise bu satır `inputSeq[0], ..., inputSeq[n-1]`'i içerir.