



تله کابین

تله کابین مائو-کونگ یک جاذبه‌ی گردشگری مشهور در تایپه است. این تله کابین از یک ریل دایره‌ای، یک ایستگاه، و n کابین تشکیل شده است که به ترتیب از ۱ تا n شماره‌گذاری شده‌اند و در یک جهت ثابت روی ریل در حال گردش هستند. در ابتدای کار، جهت حرکت به این صورت است که وقتی کابین i از ایستگاه گذشت، کابین بعدی $i + 1$ خواهد بود مگر این که $i = n$ که در این حالت کابین بعدی ۱ خواهد بود.

کابین‌ها ممکن است خراب شوند. خوشبختانه ما به تعداد نامتناهی کابین یدکی داریم که با شماره‌های $1, n+1, n+2$ و الی آخر شماره‌گذاری شده‌اند. وقتی یک کابین خراب می‌شود، ما آن را (در همان موقعیت از ریل) با اولین کابین یدکی موجود (کابین با کم‌ترین شماره) جایگزین می‌کنیم. به‌عنوان مثال، اگر ۵ کابین وجود داشته باشد و کابین ۱ خراب شود، ما آن را با کابین ۶ جایگزین می‌کنیم.

شما دوست دارید که در ایستگاه بایستید و گذشتن کابین‌ها را تماشا کنید. یک دنباله‌ی کابین، دنباله‌ای از n شماره‌ی کابین است که از ایستگاه می‌گذرند. ممکن است قبل از این که شما برسید، تعدادی از کابین‌ها خراب و جایگزین شده باشند، اما هنگامی که شما در حال تماشای کابین‌ها هستید هیچ کابینی خراب نخواهد شد.

توجه داشته باشید یک وضعیت کابین‌های روی ریل می‌تواند چندین دنباله‌ی کابین تولید کند، بسته به این که چه کابینی ابتدا از مقابل شما عبور کند. به‌عنوان مثال اگر هیچ یک از کابین‌ها خراب نشده باشد، هر دو دنباله‌ی کابین $(1, 2, 3, 4, 5)$ و $(3, 2, 1, 4, 5)$ ممکن است تولید شوند، ولی $(1, 2, 3, 4, 5)$ ممکن نیست (چرا که کابین‌ها به ترتیب نادرست ظاهر شده‌اند).

اگر کابین ۱ خراب شود، ممکن است دنباله‌ی کابین $(3, 2, 4, 5, 6)$ را مشاهده کنیم. اگر بعد از آن کابین ۴ خراب شود، آن را با کابین ۷ جایگزین می‌کنیم و ممکن است دنباله‌ی $(5, 7, 3, 2, 6)$ را مشاهده کنیم. اگر بعد از آن کابین ۷ خراب شود، آن را با کابین ۸ جایگزین می‌کنیم و ممکن است دنباله‌ی $(2, 6, 5, 8, 3)$ را مشاهده کنیم.

broken gondola	new gondola	possible gondola sequence
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

یک دنباله‌ی جایگزینی، دنباله‌ای از شماره‌ی کابین‌های خراب‌شده به ترتیب خراب شدن آن‌ها می‌باشد. در مثال قبلی، دنباله‌ی جایگزینی $(1, 4, 7)$ است. می‌گوییم دنباله‌ی جایگزینی r ، دنباله‌ی کابین g را تولید می‌کند، اگر بعد از خراب شدن کابین‌ها به ترتیب دنباله‌ی جایگزینی r ، مشاهده‌ی دنباله‌ی کابین g ممکن باشد.

بررسی دنباله‌ی کابین

در سه زیرمسئله‌ی اول شما باید بررسی کنید که آیا دنباله‌ی ورودی یک دنباله‌ی کابین است یا خیر. در جدول زیر چند نمونه دنباله را مشاهده می‌کنید که بعضی از آن‌ها دنباله‌ی کابین هستند و بعضی دیگر نیستند. شما باید تابع `valid` را پیاده‌سازی کنید.

- n : طول دنباله ورودی.
- `inputSeq`: آرایه‌ای به طول n ؛ `inputSeq[i]` عنصر i ام دنباله‌ی ورودی را نشان می‌دهد.
- تابع در صورتی که دنباله‌ی ورودی یک دنباله‌ی کابین است باید مقدار ۱ و در غیر این صورت مقدار ۰ را برگرداند.

زیرمسئله‌های ۱، ۲ و ۳

subtask	points	n	<code>inputSeq</code>
1	5	$n \leq 100$	has each number from 1 to n exactly once
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

مثال‌ها

subtask	<code>inputSeq</code>	return value	note
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 cannot appear just before 5
1	(4, 3, 2, 1)	0	4 cannot appear just before 3
2	(1, 2, 3, 4, 5, 6, 5)	0	two gondolas numbered 5
3	(2, 3, 4, 9, 6, 7, 1)	1	replacement sequence (5, 8)
3	(10, 4, 3, 11, 12)	0	4 cannot appear just before 3

دنباله‌ی جایگزینی

در سه زیرمسئله‌ی بعدی شما باید یک دنباله‌ی جایگزینی بسازید که دنباله‌ی کابین داده‌شده را تولید می‌کند. هر دنباله‌ی جایگزینی با این خاصیت پذیرفته می‌شود. شما باید تابع `replacement` را پیاده‌سازی کنید.

• `replacement(n, gondolaSeq, replacementSeq)`

- n : طول دنباله‌ی کابین داده‌شده است.
- `gondolaSeq`: آرایه‌ای به طول n ؛ می‌دانیم که `gondolaSeq` حتماً یک دنباله‌ی کابین است، و `gondolaSeq[i]` عنصر i ام دنباله است (برای $0 \leq i \leq n - 1$).
- این تابع باید مقدار l ، طول دنباله‌ی جایگزینی، را برگرداند.
- `replacementSeq`: آرایه‌ای که به اندازه‌ی کافی بزرگ است تا دنباله‌ی جایگزینی را ذخیره کند. شما باید دنباله‌ی جایگزینی خود را با گذاشتن عنصر i ام آن در درایه‌ی `replacementSeq[i]` برگردانید ($0 \leq i \leq l - 1$).

زیرمسئله‌های ۴، ۵ و ۶

subtask	points	n	<code>gondolaSeq</code>
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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

مثال‌ها

subtask	<code>gondolaSeq</code>	return value	<code>replacementSeq</code>
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	()
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

شمردن دنباله‌های جایگزینی

در چهار زیرمسئله‌ی بعد، شما باید تعداد دنباله‌های جایگزینی ممکن که یک دنباله‌ی داده‌شده را تولید می‌کنند، به پیمانه‌ی $1,000,000,009$ بیابید (دنباله‌ی داده‌شده می‌تواند یک دنباله‌ی کابین باشد یا نباشد). برای این منظور شما باید تابع `countReplacement` را پیاده‌سازی کنید.

`countReplacement(n, inputSeq)` •

- n : طول دنباله‌ی ورودی.
- `inputSeq`: آرایه به طول n ; `inputSeq[i]` عنصر i از دنباله‌ی ورودی است ($0 \leq i \leq n - 1$).
- اگر دنباله‌ی ورودی یک دنباله‌ی کابین باشد، تعداد دنباله‌های جایگزینی را بیابید که این دنباله‌ی کابین را تولید می‌کنند (که این تعداد ممکن است خیلی بزرگ باشد)، و مقدار آن را به پیمانه‌ی $1,000,000,009$ برگردانید. اگر دنباله‌ی ورودی یک دنباله‌ی کابین نباشد، این تابع باید مقدار 0 را برگرداند. اگر دنباله‌ی ورودی یک دنباله‌ی کابین باشد، اما هیچ کابینی خراب نشده باشد، تابع باید مقدار 1 را برگرداند.

زیرمسئله‌های ۷، ۸، ۹ و ۱۰

subtask	points	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, and at least $n - 3$ of the initial gondolas $1, \dots, n$ did not break down.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

مثال‌ها

subtask	inputSeq	return value	replacement sequence
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 is not a gondola sequence
10	(3, 4)	2	(1, 2) or (2, 1)

جزئیات پیاده‌سازی

شما باید دقیقاً یک فایل با نام `gondola.c`، `gondola.cpp` یا `gondola.pas` ارسال کنید. این فایل باید هر سه زیربرنامه‌ای که در بالا توضیح داده شد را به صورت زیر پیاده‌سازی کند (حتی اگر قصد دارید فقط بعضی از زیرمسئله‌ها را حل کنید). همچنین شما باید هدر فایل `gondola.h` را در برنامه‌ی C/C++ خود درج کنید.

برنامه‌ی C/C++

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

برنامه‌ی پاسکال

```
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;
```

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

- خط ۱: T ، شماره‌ی زیرمسئله‌ای که برنامه‌ی شما قرار است حل کند ($1 \leq T \leq 10$).
- خط ۲: n ، طول دنباله‌ی ورودی.
- خط ۳: اگر T برابر ۴، ۵، یا ۶ باشد، این خط شامل `gondolaSeq[0]`، ...، `gondolaSeq[n-1]` خواهد بود. در غیر این صورت این خط شامل `inputSeq[0]`، ...، `inputSeq[n-1]` می‌باشد.