



## غوندولا

يعد ماو-كونغ غوندولا معلماً هاماً في تايبيه. يتألف نظام غوندولا من سكة دائرية، ومحطة واحدة  $n$  غوندولا مرقمة بشكل متسلسل من 1 وحتى  $n$  تدور على السكة في اتجاه واحد.

في البداية بعد أن يعبر الغوندولا رقم  $i$  المحطة، فإن الغوندولا التالي الذي سيعبر المحطة هو الغوندولا رقم  $i + 1$  إذا كان  $i < n$ ، أو الغوندولا رقم 1 إذا كان  $i = n$ .

يمكن للغوندولا أن يتعطل. ولحسن الحظ يمكننا الحصول على عدد لا نهائي من الغوندولا، والتي ترقم  $n + 1$ ,  $n + 2$ , وهكذا، عندما يتعطل أحد الغوندولا نقوم باستبداله (بنفس المكان على المسار) بأول غوندولا متاح أي بذلك الذي يحمل أقل رقم ممكن. مثلاً: إذا كان هناك خمس غوندولا وتعطل الغوندولا رقم 1، عندئذ سنقوم باستبداله بالغوندولا رقم 6.

أنت تحب أن تقف عند المحطة لتشاهد الغوندولا وهي تمر بالمحطة، يعرف تسلسل الغوندولا بأنه سلسلة من  $n$  ارقام الغوندولا التي تعبر المحطة، من الممكن أن يتعطل غوندولا واحد أو أكثر (وقد تم تبديلها) قبل أن تحضر أنت للمحطة، ولكن لا تتعطل الغوندولا خلال فترة مشاهدتك.

لاحظ أنه من الممكن أن يعطي نفس ترتيب الغوندولا على السكة عدة تسلسلات غوندولا، وذلك يعتمد على الغوندولا الذي سيمر أولاً بعد لحظة وصولك إلى المحطة. مثلاً إذا لم تعطل أي غوندولا عندئذ يكون كل من (2, 3, 4, 5, 1) و (4, 5, 1, 2, 3) احتمالان لسلسلة غوندولا ولكن (4, 3, 2, 5, 1) ليست احتمالاً ممكناً (لأن الغوندولا يظهر بترتيب خاطئ).

إذا تعطل الغوندولا رقم 1، يمكننا أن نشاهد تسلسل الغوندولا (4, 5, 6, 2, 3). ثم إذا تعطل الغوندولا رقم 4، سنقوم باستبداله بالغوندولا رقم 7 ويمكن أن نشاهد تسلسل الغوندولا (6, 2, 3, 7, 5). إذا تعطل الغوندولا رقم 7 بعد ذلك، سنقوم باستبداله بالغوندولا رقم 8 ويمكننا الآن الحصول على تسلسل الغوندولا (3, 8, 5, 6, 2).

تسلسل غوندولا ممكن	الغوندولا الجديد	الغوندولا المتعطل
(4, 5, 6, 2, 3)	6	1
(6, 2, 3, 7, 5)	7	4
(3, 8, 5, 6, 2)	8	7

يعرف تسلسل الاستبدال بأنه تسلسل يتألف من ارقام الغوندولا التي تعطلت، بالترتيب الذي تعطلت به. مثلاً في المثال السابق، فإن تسلسل الاستبدال هو (1, 4, 7). يؤدي تسلسل الاستبدال  $r$  لإنتاج تسلسل غوندولا  $g$  إذا كان بعد تعطل الغوندولا وفقاً لتسلسل الاستبدال  $r$ ، يكون تسلسل الغوندولا  $g$  أحد التسلسلات الممكنة.

## التحقق من تسلسل غوندولا

في أول ثلاث مهام جزئية يجب عليك التأكد فيما إذا كان تسلسل معين مدخل هو تسلسل غوندولا. انظر الجدول أدناه لأمثلة على تسلسلات غوندولا صالحة وغير صالحة، يجب عليك كتابة التابع `valid`.

■ `valid(n, inputSeq)`

■ `n`: طول تسلسل الدخل.

■ `inputSeq`: مصفوفة طوله `n`; `inputSeq[i]` هو العنصر رقم  $i$  من تسلسل الدخل، من أجل  $0 \leq i \leq n - 1$ .

■ يجب على التابع أن يعيد 1 إذا كانت سلسلة الدخل هي سلسلة غوندولا وإلا يعيد 0.

### المسائل الجزئية 1, 2, 3

المسألة الجزئية	النقاط	$n$	inputSeq
1	5	$n \leq 100$	تحتوي الأرقام من 1 حتى $n$ تماماً مرة واحدة
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

أمثلة

المسألة الجزئية	inputSeq	القيمة المعادة	ملاحظات
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	لا يمكن ل 1 أن تأتي تماماً قبل 5
1	(4, 3, 2, 1)	0	لا يمكن ل 4 أن تأتي تماماً قبل 3
2	(1, 2, 3, 4, 5, 6, 5)	0	هناك غوندولا اثنان بنفس الرقم 5
3	(2, 3, 4, 9, 6, 7, 1)	1	(تسلسل الاستبدال (5, 8)
3	(10, 4, 3, 11, 12)	0	لا يمكن ل 4 أن تأتي تماماً قبل 3

### تسلسل الاستبدال

في المسائل الجزئية الثلاثة التالية يجب عليك بناء تسلسل الاستبدال الذي يؤدي إلى الحصول على تسلسل غوندولا معطى. سيتم قبول أي تسلسل استبدال إن وجد أكثر من واحد، عليك أن تحقق التابع replacement.

■ replacement(n, gondolaSeq, replacementSeq)

■  $n$  هو طول تسلسل غوندولا.

■ gondolaSeq: مصفوفة طولها  $n$ ; gondolaSeq حيث أنه من المضمون أنها تمثل تسلسل غوندولا صالحاً، و [gondolaSeq[i] هو العنصر رقم  $i$  من التسلسل، for  $0 \leq i \leq n - 1$ .

■ يجب على التابع إعادة  $l$ ، وهو طول سلسلة الاستبدال.

■ replacementSeq: مصفوفة كبيرة بشكل كافي لتخزين تسلسل الاستبدال; يجب عليك إعادة تسلسل عن طريق وضع العنصر رقم  $i$  من تسلسل الاستبدال الخاص بك في replacementSeq[i], من أجل  $0 \leq i \leq l - 1$ .

## المسائل الجزئية 4, 5, 6

المسألة الجزئية	النقاط	$n$	<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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

أمثلة

المسألة الجزئية	<code>gondolaSeq</code>	القيمة المعادة	<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)

## عدد تسلسلات الاستبعاد

في المسائل الجزئية الأربعة التالية يجب عليك عد عدد تسلسلات الاستبعاد الممكنة والتي تعطي تسلسلاً معطى (والذي يمكن أن يكون تسلسل غوندولا ويمكن أن لا يكون)، يجب عليك الحصول على باقي قسمة هذا العدد على  $1,000,000,009$ . يجب عليك تحقيق التابع `countReplacement`.

■ `countReplacement(n, inputSeq`

■ `n`: طول تسلسل الدخل.

■ `inputSeq`: مصفوفة طولها  $n$ ; `inputSeq[i]` هو العنصر  $i$  من تسلسل الدخل، من أجل  $0 \leq i \leq n - 1$ .

■ إذا كان تسلسل الدخل هو تسلسل غوندولا، يجب عليك عد عدد تسلسلات الاستبدال التي تنتج تسلسل غوندولا هذا (وهذا العدد ممكن أن يكون كبيراً جداً) ويجب عليك إعادة باقي قسمة هذا العدد على  $1,000,000,009$ . أما إذا كان تسلسل الدخل ليس تسلسل غوندولان يجب على التابع أن يعيد 0، إذا كان تسلسل الدخل هو سلسلة غوندولا ولكن بدون غوندولا متعطلة يجب أن يعيد التابع القيمة 1.

## المسائل الجزئية 7, 8, 9, 10

المسألة الجزئية	النقاط	$n$	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	وعلى الأقل $n - 3$ من الغوندولا الأولية، $1 \leq \text{inputSeq}[i] \leq 100$ ، $1, \dots, n$ أم تتعطل.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

المسألة الجزئية	inputSeq	القيمة المعادة	تسلسل الاستبدال
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	ليس تسلسل غوندولا inputSeq
10	(3, 4)	2	(1, 2) or (2, 1)

## تفاصيل التنجيز

يجب عليك إرسال ملف واحد فقط لا أكثر ولا أقل اسمه gondola.c, gondola.cpp أو gondola.pas. يجب على هذا الملف أن يحقق كل البرامج الجزئية الموضحة أعلاه (حتى ولو كنت تنوي حل بعض المهمات الجزئية)، باستخدام ترويسات التوابع التالية، كما يجب عليك تضمين ملف الترويسات gondola.h من أجل ++C/C.

### C/C++ programs

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

### Pascal programs

```
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;
```

### المصحح التجريبي

يقرأ المصحح التجريبي الدخل بالتنسيق التالي:

- (1 ≤ T ≤ 10) line 1: T, the subtask number your program intends to solve
- line 2: n, the length of the input sequence
- line 3: If T is 4, 5, or 6, this line contains gondolaSeq[0], ..., gondolaSeq[n-1]. Otherwise . [this line contains inputSeq[0], ..., inputSeq[n-1]