



Gondola

A gôndola (teleférico) Mao-Kong é uma atração famosa de Taipei. O sistema da gôndola consiste de um trilho circular, uma única estação, e n gôndolas numeradas consecutivamente de 1 até n movimentando-se ao redor do trilho em uma direção fixa. Após a gôndola i passar pela estação, a próxima gôndola a passar pela estação será a gôndola $i + 1$ se $i < n$, ou a gôndola 1 se $i = n$.

Gôndolas podem quebrar. Felizmente temos um suprimento infinito de gôndolas sobressalentes, que são numeradas $n + 1$, $n + 2$, e assim por diante. Quando uma gôndola quebra nós a substituímos (na mesma posição do trilho) pela primeira gôndola sobressalente disponível, isto é, a que tem o menor número. Por exemplo, se há cinco gôndolas e a gôndola 1 quebra, então nós vamos substituí-la pela gôndola 6.

Você gosta de ficar na estação e ver as gôndolas passarem. Uma *sequência de gôndolas* é uma sequência de n números de gôndolas que passam pela estação. Uma ou mais gôndolas podem ter quebrado (e sido substituídas) antes de você chegar, mas nenhuma das gôndolas quebra enquanto você está olhando.

Note que a mesma configuração de gôndolas no trilho pode produzir múltiplas sequências de gôndolas, dependendo de qual gôndola passa primeiro quando você chega na estação. Por exemplo, se nenhuma gôndola tiver quebrado então tanto (2, 3, 4, 5, 1) quanto (4, 5, 1, 2, 3) são possíveis sequências de gôndolas, mas (4, 3, 2, 5, 1) não é (porque as gôndolas aparecem na ordem errada).

Se a gôndola 1 quebrar, então poderemos observar a sequência de gôndolas (4, 5, 6, 2, 3). Se a gôndola 4 quebrar em seguida, nós a substituímos pela gôndola 7 e poderemos observar a sequência de gôndolas (6, 2, 3, 7, 5). Se a gôndola 7 quebrar após isso, nós a substituímos pela gôndola 8 e poderemos observar a sequência de gôndolas (3, 8, 5, 6, 2).

gôndola que brada	nova gôndola	possível sequência de gôndolas
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

Uma *sequência de substituição* é uma sequência consistindo dos números das gôndolas que quebraram, na ordem em que quebraram. No exemplo anterior a sequência de substituição é (1, 4, 7). Uma sequência de substituição r produz uma sequência de gôndolas g se, após as gôndolas quebrarem de acordo com a sequência de substituição r , a sequência de gôndolas g puder ser observada.

Verificação de Sequências de Gôndolas

Nas primeiras três subtarefas você precisa verificar se uma sequência da entrada é uma sequência de gôndolas. Veja a tabela abaixo para exemplos de sequências que são ou que não são sequências de

gôndolas. Você precisa implementar a função `valid`.

- `valid(n, inputSeq)`
- n : o tamanho da sequência da entrada.
- `inputSeq`: vetor de tamanho n ; `inputSeq[i]` é o elemento i da sequência da entrada, para $0 \leq i \leq n - 1$.
- A função deve retornar 1 se a sequência da entrada for uma sequência de gôndolas, ou 0 caso contrário.

Subtarefas 1, 2, 3

subtarefa	pontos	n	<code>inputSeq</code>
1	5	$n \leq 100$	contém cada número de 1 até n exatamente uma vez
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

Exemplos

subtarefa	<code>inputSeq</code>	valor de retorno	nota
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 não pode aparecer imediatamente antes do 5
1	(4, 3, 2, 1)	0	4 não pode aparecer imediatamente antes do 3
2	(1, 2, 3, 4, 5, 6, 5)	0	duas gôndolas de número 5
3	(2, 3, 4, 9, 6, 7, 1)	1	sequência de substituição (5, 8)
3	(10, 4, 3, 11, 12)	0	4 não pode aparecer imediatamente antes do 3

Sequência de Substituição

Nas próximas três subtarefas você deve construir uma sequência de substituição possível que produz a sequência de gôndola da entrada. Qualquer sequência de substituição assim será aceita. Você precisa implementar a função `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
- n é o tamanho da sequência de gôndolas.
- `gondolaSeq`: vetor de tamanho n ; `gondolaSeq` é garantidamente uma sequência de gôndolas, e `gondolaSeq[i]` é o elemento i da sequência, para $0 \leq i \leq n - 1$.
- A função deve retornar l , o tamanho da sequência de substituição.

- `replacementSeq`: vetor grande o suficiente para guardar a sequência de substituição; você deve retornar sua sequência colocando o elemento i de sua sequência de substituição em `replacementSeq[i]`, para $0 \leq i \leq l - 1$.

Subtarefas 4, 5, 6

subtarefa	pontos	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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

Exemplos

subtarefa	<code>gondolaSeq</code>	valor de retorno	<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)

Contar Sequências de Substituições

Nas próximas quatro subtarefas você deve contar o número de possíveis sequências de substituição que produzem uma sequência da entrada (que não necessariamente será uma sequência de gôndolas), modulo **1,000,000,009**. Você precisa implementar a função `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : o tamanho da sequência da entrada.
 - `inputSeq`: vetor de tamanho n ; `inputSeq[i]` é o elemento i da sequência da entrada, para $0 \leq i \leq n - 1$.
 - Se a sequência da entrada for uma sequência de gôndolas, então conte o número de sequências de substituição que produzem esta sequência de gôndolas (este número pode ser muito grande), e retorne este número modulo **1,000,000,009**. Se a sequência da entrada não for uma sequência de gôndolas, a função deve retornar 0. Se a sequência da entrada for uma sequência de gôndola mas nenhuma gôndola quebrou, a função deve retornar 1.

Subtarefas 7, 8, 9, 10

subtarefas	pontos	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$, e pelo menos $n - 3$ das gôndolas iniciais $1, \dots, n$ não quebraram.

subtarefas	pontos	n	inputSeq
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

Exemplos

subtarefas	inputSeq	valor de retorno	sequência de substituição
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 não é uma sequência de gôndolas
10	(3, 4)	2	(1, 2) or (2, 1)

Detalhes de implementação

Você deve submeter exatamente um arquivo, chamado `gondola.c`, `gondola.cpp` ou `gondola.pas`. Este arquivo deve implementar todos os três subprogramas descritos acima (mesmo se você planeja resolver apenas algumas das subtarefas), usando as seguintes assinaturas. Você também precisa incluir o arquivo de cabeçalho `gondola.h` para implementação C/C++.

Programas em C/C++

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

Programas em Pascal

```
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;
```

Avaliador exemplo

O avaliador exemplo lê da entrada no seguinte formato:

- linha 1: T , o número da subarefa que seu programa pretende resolver ($1 \leq T \leq 10$).
- linha 2: n , o tamanho da sequência de entrada.
- linha 3: Se T for 4, 5, ou 6, esta linha contém `gondolaSeq[0], ..., gondolaSeq[n-1]`. Caso contrário esta linha contém `inputSeq[0], ..., inputSeq[n-1]`.