



# გონდოლა

"მაო-კონგ გონდოლა" წარმოადგენს ტაივანის ერთ-ერთ ღირსშესანიშნაობას. გონდოლათა სისტემა შეიცავს წრიულ გზას, ერთადერთ სადგურს და  $n$  რაოდენობის გონდოლას, გადანომრილს მიმდევრობით 1-დან  $n$ -მდე. გონდოლები მოძრაობენ ფიქსირებული მიმართულებით. თავდაპირველად, სადგურზე  $i$ -ური გონდოლის გავლის შემდეგ გამოჩნდება  $i + 1$  ნომრის მქონე გონდოლა, თუკი  $i < n$ , ან გამოჩნდება გონდოლა ნომრით 1, თუ  $i = n$ .

გონდოლები შეიძლება დაზიანდნენ. საბედნიეროდ, ჩვენ გვაქვს სათადარიგო გონდოლების შეუზღუდავი რაოდენობა, გადანომრილი  $n + 1$ ,  $n + 2$ , და ა.შ. როცა გონდოლა დაზიანდება, ჩვენ ვცვლით მას (ზუსტად იმავე პოზიციაში) პირველი ხელმისაწვდომი სათადარიგო გონდოლით, რომელსაც აქვს უმცირესი ნომერი. მაგალითად, თუ ჩვენ გვაქვს 5 გონდოლა და დაზიანდა პირველი, ჩვენ მას შევცვლით გონდოლით, რომლის ნომერი იქნება 6.

თქვენ გიყვართ სადგურზე დგომა და დაკვირვება გონდოლათა მოძრაობაზე. "გონდოლათა მიმდევრობა" წარმოადგენს  $n$  ცალი რიცხვისაგან შედგენილ გონდოლათა ნომრების მიმდევრობას. თქვენს მისვლამდე შეიძლება ერთი ან რამდენიმე გონდოლა დაზიანდა (და შესაბამისად შეიცვალა კიდეც), მაგრამ თქვენი დაკვირვების განმავლობაში გონდოლის დაზიანება აღარ ხდება.

შევნიშნოთ, რომ გონდოლათა მიმდევრობა შეიძლება სხვადასხვა იყოს იმის მიხედვით, თუ თქვენ როდის მიხვედით სადგურზე. მაგალითად, თუკი არცერთი გონდოლა არ დაზიანებულა, მიმდევრობა შეიძლება იყოს (2, 3, 4, 5, 1) ან (4, 5, 1, 2, 3), მაგრამ არ შეიძლება იყოს (4, 3, 2, 5, 1), რადგან გონდოლების თანმიმდევრობა არასწორია.

თუ დაზიანდა გონდოლა ნომრით 1, ჩვენ შეიძლება მივიღოთ მიმდევრობა (4, 5, 6, 2, 3). თუკი ამის შემდეგ დაზიანდა გონდოლა ნომრით 4, იგი შეიცვლება 7 ნომრის მქონე გონდოლით და მივიღებთ მიმდევრობას (6, 2, 3, 7, 5). თუ ამის შემდეგ დაზიანდება 7 ნომრის მქონე გონდოლა, იგი შეიცვლება მე-8 გონდოლით და მივიღებთ მიმდევრობას (3, 8, 5, 6, 2).

დაზიანებული გონდოლა	ახალი გონდოლა	გონდოლათა შესაძლო მიმდევრობა
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

ცვლილებათა მიმდევრობა შეიცავს დაზიანებული გონდოლების ნომრებს შესაბამისი თანმიმდევრობით. ზემოთ განხილულ მაგალითში ცვლილებათა მიმდევრობაა (1, 4, 7). ცვლილებათა  $r$  მიმდევრობა *ჰქმნის* გონდოლათა  $g$

მიმდევრობას, თუ ცვლილებათა  $r$  მიმდევრობაში მითითებული გონდოლების დაზიანების შემდეგ, მიიღება გონდოლათა  $g$  მიმდევრობა.

## გონდოლათა მიმდევრობის შემოწმება

პირველ 3 ქვეამოცანაში თქვენ უნდა დაადგინოთ, შემოსატანი მიმდევრობა წარმოადგენს თუ არა გონდოლათა მიმდევრობას. ქვემოთ იხილეთ მიმდევრობათა ცხრილი და განსაზღვრეთ, წარმოადგენს თუ არ ისინი გონდოლათა მიმდევრობას. ამისათვის თქვენ უნდა განახორციელოთ ფუნქცია `valid`-ის იმპლემენტაცია.

- `valid(n, inputSeq)`
  - $n$ : შემოსატანი მიმდევრობის სიგრძე.
  - `inputSeq`:  $n$  სიგრძის მქონე მასივი; `inputSeq[i]` არის შემოსატანი მიმდევრობის  $i$ -ური ელემენტი, ნებისმიერი  $0 \leq i \leq n - 1$ -სათვის.
  - ფუნქციამ უნდა დააბრუნოს 1, თუ შემოსატანი მიმდევრობა წარმოადგენს გონდოლათა მიმდევრობას, ხოლო წინააღდეგ შემთხვევაში გამოიტანოს 0.

### ქვეამოცანები 1, 2, 3

ქვეამოცანები	ქულა	$n$	<code>inputSeq</code>
1	5	$n \leq 100$	შეიცავს რიცხვებს მხოლოდ 1-დან $n$ -მდე დიაპაზონში
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

### მაგალითები

ქვეამოცანა	<code>inputSeq</code>	return მნიშვნელობა	შენიშვნა
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 არ შეიძლება იყოს 5-ის წინ
1	(4, 3, 2, 1)	0	4 არ შეიძლება იყოს 3-ის წინ
2	(1, 2, 3, 4, 5, 6, 5)	0	ორ გონდოლას აქვს ნომერი 5
3	(2, 3, 4, 9, 6, 7, 1)	1	ცვლილებათა მიმდევრობაა (5, 8)

ქვეამოცანა	inputSeq	return მნიშვნელობა	შენიშვნა
3	(10, 4, 3, 11, 12)	0	4 არ შეიძლება იყოს 3-ის წინ

## ცვლილებათა მიმდევრობა

მომდევნო სამ ქვეამოცანაში თქვენ უნდა ააგოთ ცვლილებათა შესაძლო მიმდევრობა, რომელიც შეჰქმნის გონდოლათა მოცემულ მიმდევრობას. გონდოლათა ნებისმიერი მიმდევრობის მიღება შესაძლებელია. თქვენ უნდა მოახდინოთ ფუნქცია replacement-ის იმპლემენტაცია.

- replacement(*n*, gondolaSeq, replacementSeq)
  - *n* წარმოადგენს გონდოლათა მიმდევრობის სიგრძეს.
  - gondolaSeq: *n* სიგრძის მქონე მასივი; gondolaSeq გარანტირებულად არის გონდოლათა მიმდევრობა და gondolaSeq[i] არის მიმდევრობის *i*-ური ელემენტი, ნებისმიერი ( $0 \leq i \leq n - 1$ )-თვის.
  - ფუნქციამ უნდა დააბრუნოს *l* - ცვლილებათა მიმდევრობის სიგრძე.
  - replacementSeq: მასივი, რომელიც საკმარისად დიდია საიმისოდ, რომ შეინახოს ცვლილებათა მიმდევრობა; თქვენ უნდა დააბრუნოთ თქვენი მიმდევრობა, სადაც *i*-ური ელემენტი მოთავსებული იქნება replacementSeq[i]-ში, ნებისმიერი ( $0 \leq i \leq l - 1$ )-თვის.

## ქვეამოცანები 4, 5, 6

ქვეამოცანა	ქულები	<i>n</i>	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	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

## მაგალითები

ქვეამოცანა	gondolaSeq	return მნიშვნელობა	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)

## ცვლილებათა მიმდევრობების დათვლა

მომდევნო ოთხ ქვეამოცანაში თქვენ უნდა გამოთვალოთ ცვლილებათა იმ შესაძლო მიმდევრობების რაოდენობა, რომლებიც ჰქმნიან მოცემულ

მიმდევრობას (რომელიც შეიძლება იყოს და შეიძლება არ იყოს გონდოლათა მიმდევრობა). თქვენ უნდა გამოიტანოთ მიღებული პასუხის 1,000,000,009-ზე მოდულით განაყოფი. მოახდინეთ ფუნქცია countReplacement-ის იმპლემენტაცია.

- countReplacement(n, inputSeq)
  - n: შესატანი მიმდევრობის სიგრძე.
  - inputSeq: n სიგრძის მასივი; inputSeq[i] არის შესატანი მიმდევრობის i-ური ელემენტი, ნებისმიერი ( $0 \leq i \leq n - 1$ )-თვის.
  - თუ შესატანი მიმდევრობა არის გონდოლათა მიმდევრობა, თქვენ უნდა გამოთვალოთ იმ ცვლილებათა მიმდევრობების რაოდენობა, რომლებიც ჰქმნიან გონდოლათა მოცემულ მიმდევრობას. რადგან მიღებული პასუხი შეიძლება ძალიან დიდი იყოს, გამოიტანეთ მისი 1,000,000,009-ზე მოდულით განაყოფი. თუ შესატანი მიმდევრობა არ წარმოადგენს გონდოლათა მიმდევრობას, ფუნქციამ უნდა დააბრუნოს 0. თუ შესატანი მიმდევრობა წარმოადგენს გონდოლათა მიმდევრობას, მაგრამ არცერთი გონდოლა არ დაზიანებულია – გამოიტანეთ 1.

## ქვეამოცანები 7, 8, 9, 10

ქვეამოცანა	ქულა	n	inputSeq
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ , და არანაკლებ $n - 3$ გონდოლა დაუზიანებელია საწყისი $1, \dots, n$ მიმდევრობიდან.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

## Examples

ქვეამოცანა	inputSeq	return მნიშვნელობა	ცვლილებათა მიმდევრობა
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 არ არის გონდოლათა მიმდევრობა
10	(3, 4)	2	(1, 2) or (2, 1)

## იმპლემენტაციის დეტალები

თქვენ უნდა წარმოადგინოთ ზუსტად ერთი ფაილი, რომელსაც ერქმევა

gondola.c, gondola.cpp ან gondola.pas. ეს ფაილი უნდა ახდენდეს ზემოთ აღწერილი ფაილების იმპლემენტაციას მითითებული შაბლონების გამოყენებით. C/C++ პროგრამული ენის შემთხვევაში გამოიყენეთ ფაილი gondola.h.

## 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;
```

## სანიმუშო გრადერი

შესატანი მონაცემები გრადერმა უნდა წაიკითხოს შემდეგი ფორმატით:

- სტრიქონი 1:  $T$ , იმ ქვეამოცანის ნომერი, რომელსაც ხსნის თქვენი პროგრამა ( $1 \leq T \leq 10$ ).
- სტრიქონი 2:  $n$ , შესატანი მიმდევრობის სიგრძე.
- სტრიქონი 3: თუ  $T$  არის 4, 5 ან 6, მაშინ სტრიქონი შეიცავს  $inputSeq[0], \dots, inputSeq[n-1]$ . სხვა შემთხვევაში სტრიქონი შეიცავს  $gondolaSeq[0], \dots, gondolaSeq[n-1]$ .