



Köisraudtee

Mao-Kongi köisraudtee on üks Taipei kuulsamaid atraktsioone. Sellel raudteel on üks ringikujuline köis, üks jaam ja n gondlit, mis on nummerdatud $1, \dots, n$ nende köiel rippumise järjekorras. See tähendab, et pärast gondli i möödumist jaamast tuleb sinna järgmisena gondel $i + 1$, kui $i < n$, või gondel 1 , kui $i = n$.

Gondlid võivad vahel ka rikki minna. Õnneks on rikkis gondlite asendamiseks piiramatus koguses varugondleid, mis on nummerdatud $n + 1, n + 2$ jne. Kui mõni gondel rikki läheb, asendatakse ta (samal kohal köiel) järgmise saadaoleva (st vähima veel kasutamata numbriga) varugondliga. Näiteks, kui köiel on viis gondlit ja gondel 1 läheb rikki, pannakse tema asemele gondel 6.

Sulle meeldib jaamas seista ja mööduvaid gondleid vaadata. *Gondlite jada* on n arvust koosnev jada, mis näitab gondlite numbreid nende jaamast möödumise järjekorras. On võimalik, et enne Sinu saabumist on mõned gondlid rikki läinud (ja varugondlitega asendatud), aga võib eeldada, et sel ajal kui Sa gondleid vaatad, ükski neist rikki ei lähe.

Pane tähele, et ühele gondlite konfiguratsioonile võib vastata mitu võimalikku gondlite jada, sõltuvalt sellest, milline gondel möödub esimesena pärast Sinu jaama saabumist. Näiteks, kui ükski gondel pole katki läinud, on jaded $(2, 3, 4, 5, 1)$ ja $(4, 5, 1, 2, 3)$ mõlemad võimalikud, aga jada $(4, 3, 2, 5, 1)$ pole võimalik (sest gondlid ilmuvad jaama vales järjekorras).

Kui gondel 1 läheb rikki, võime saada näiteks jada $(4, 5, 6, 2, 3)$. Kui järgmisena läheb rikki gondel 4, pannakse selle asemele gondel 7 ja siis võime saada näiteks jada $(6, 2, 3, 7, 5)$. Kui seejärel gondel 7 rikki läheb, pannakse selle asemele gondel 8 ja siis võime saada näiteks jada $(3, 8, 5, 6, 2)$.

rikkis gondel	uus gondel	võimalik jada
1	6	$(4, 5, 6, 2, 3)$
4	7	$(6, 2, 3, 7, 5)$
7	8	$(3, 8, 5, 6, 2)$

Asenduste jada on jada, mis koosneb rikki läinud gondlite numbritest nende rikkimise järjekorras. Eelmisele näitele vastav asenduste jada on seega $(1, 4, 7)$. Ütleme, et asenduste jada r tekitab gondlite jada g , kui pärast gondlite rikkimineku jada r järjekorras on võimalik näha mööduvaid gondleid jada g järjekorras.

Gondlite jada kontrollimine

Kolmes esimeses alamülesandes tuleb kontrollida, kas antud jada on võimalik gondlite jada. Allolevas tabelis on näiteid jadadest mis on ja mis ei ole võimalikud gondlite jaded. Programmis tuleb realiseerida funktsioon `valid`.

- `valid(n, inputSeq)`
 - `n`: sisendjada pikkus.
 - `inputSeq`: n -elemendiline massiiv; `inputSeq[i]` on iga $0 \leq i \leq n - 1$ korral sisendjada i . element.
 - Funktsioon peab tagastama 1, kui sisendjada on võimalik gondlite jada, ja 0, kui ei ole.

Alamülesanded 1, 2, 3

alamülesanne	punkte	n	<code>inputSeq</code>
1	5	$n \leq 100$	sisaldab iga arvu $1, \dots, n$ täpselt ühe korra
2	5	$n \leq 100\ 000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100\ 000$	$1 \leq \text{inputSeq}[i] \leq 250\ 000$

Näited

alamülesanne	<code>inputSeq</code>	tagastatav väärtus	märkused
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 ei saa olla vahetult 5 ees
1	(4, 3, 2, 1)	0	4 ei saa olla vahetult 3 ees
2	(1, 2, 3, 4, 5, 6, 5)	0	5 esineb kaks korda
3	(2, 3, 4, 9, 6, 7, 1)	1	asenduste jada (5, 8)
3	(10, 4, 3, 11, 12)	0	4 ei saa olla vahetult 3 ees

Asenduste jada leidmine

Kolmes järgmises alamülesandes tuleb koostada asenduste jada, mis tekitab antud gondlite jada. Kui võimalikke asenduste jadasid on mitu, sobib ükskõik milline neist. Programmis tuleb realiseerida funktsioon `replacement`.

- `replacement(n, gondolaSeq, replacementSeq)`
 - `n`: gondlite jada pikkus.
 - `gondolaSeq`: n -elemendiline massiiv; võib eeldada, et `gondolaSeq` on võimalik gondlite jada ja `gondolaSeq[i]` on iga $0 \leq i \leq n - 1$ korral selle jada i . element.
 - Funktsioon peab tagastama asenduste jada pikkuse l .
 - `replacementSeq`: massiiv, mis on piisavalt suur, et asenduste jada sinna ära mahuks; funktsioon peab asenduste jada tagastamiseks panema iga $0 \leq i \leq l - 1$ korral selle jada i . elemendi massiivi elementi `replacementSeq[i]`.

Alamülesanded 4, 5, 6

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

Näited

alamülesanne	<code>gondolaSeq</code>	tagastatav väärtus	<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)

Asenduste jadade loendamine

Neljas järgmises alamülesandes tuleb leida kõigi selliste asendusjadade arv, mis tekitavad antud jada (mis võib olla või mitte olla võimalik gondlite jada), *modulo* **1 000 000 009**. Programmis tuleb realiseerida funktsioon `countReplacement`.

- `countReplacement(n, inputSeq)`
 - n : sisendjada pikkus.
 - `inputSeq`: n -elementiline massiiv; `inputSeq[i]` on iga $0 \leq i \leq n - 1$ korral sisendjada i . element.
 - Kui sisendjada on võimalik gondlite jada, peab funktsioon leidma, kui palju on asenduste jadasid, mis selle gondlite jada tekitavad, ja tagastama asenduste jadade arvu (mis võib olla väga suur) **1 000 000 009**-ga jagamisel tekkiva jäägi. Kui sisendjada ei ole võimalik gondlite jada, peab funktsioon tagastama 0. Kui sisendjada on gondlite jada, kus ükski gondel pole rikki läinud, peab funktsioon tagastama 1.

Alamülesanded 7, 8, 9, 10

alamülesanne	punkte	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ ja esialgsete gondlite $1, \dots, n$ hulgast vähemalt $n - 3$ ei ole rikki läinud
9	15	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq 250\,000$
10	10	$n \leq 100\,000$	$1 \leq \text{inputSeq}[i] \leq 1\,000\,000\,000$

Näited

alamülesanne	inputSeq	tagastatav väärtus	asenduste jada
7	(1, 2, 7, 6)	2	(3, 4, 5) või (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	inputSeq ei ole gondlite jada
10	(3, 4)	2	(1, 2) või (2, 1)

Realisatsioon

Esitada tuleb täpselt üks fail nimega `gondola.c`, `gondola.cpp` või `gondola.pas`. Selles failis peavad olema eelpool kirjeldatud funktsioonid järgnevate signatuuridega (failis peavad olema kõik kolm funktsiooni, isegi kui Sa lahendad ainult osa alamülesandeid). C ja C++ lahendused peavad lisaks kaasama päisfaili `gondola.h`.

C ja C++

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

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;
```

Näidishindaja

Näidishindaja eeldab järgmist sisendi vormingut:

- Esimesel real: alamülesande number T ($1 \leq T \leq 10$).
- Teisel real: sisendjada pikkus n .
- Kolmandal real: Kui T on 4, 5 või 6, siis `gondolaSeq[0], ..., gondolaSeq[n-1]`, muidu `inputSeq[0], ..., inputSeq[n-1]`.