



International Olympiad in Informatics 2014

13-20th July 2014

Taipei, Taiwan

Day-2 tasks

holiday

Language: fi-FI

Loma

Jian-Jia suunnittelee seuraavaa lomaansa Taiwanin. Lomansa aikana Jian-Jia liikkuu kaupungista toiseen ja vierailee niiden nähtävyyksissä.

Taiwanissa on n kaupunkia, jotka kaikki sijaitsevat saman moottoritien varrella. Kaupungit on numeroitu $0, \dots, n - 1$. Kaupungin i , jossa $0 < i < n - 1$, viereiset kaupungit ovat $i - 1$ ja $i + 1$. Ainoa kaupungin 0 viereinen kaupunki on 1 , ja ainoa kaupungin $n - 1$ viereinen kaupunki on $n - 2$.

Jokaisessa kaupungissa on jokin määrä nähtävyyksiä. Jian-Jialla on d päivän loma, ja hän aikoo vierailla mahdollisimman monessa nähtävyydessä. Jian-Jia on jo valinnut kaupungin, josta hän aloittaa lomansa. Joka päivä lomalla Jian-Jia voi joko siirtyä viereiseen kaupunkiin tai vierailla kaikissa nykyisen kaupungin nähtävyyksissä. Hän ei voi kuitenkaan tehdä molempia asioita. Jian-Jia *ei koskaan vieraile saman kaupungin nähtävyyksissä useasti*, vaikka hän kävisi kaupungissa monta kertaa. Tehtäväsi on auttaa Jian-Jiaa suunnittelemaan loma niin, että hän vierailee mahdollisimman monessa nähtävyydessä.

Esimerkki

Oletetaan, että Jian-Jian loman kesto on 7 päivää, kaupunkeja on 5 (listattu alla olevassa taulukossa) ja hän aloittaa kaupungista 2. Ensimmäisenä päivänä Jian-Jia vierailee 20 nähtävyydessä kaupungissa 2. Toisena päivänä Jian-Jia siirtyy kaupungista 2 kaupunkiin 3, ja kolmantena päivänä hän vierailee 30 nähtävyydessä kaupungissa 3. Tämän jälkeen Jian-Jia käyttää kolme päivää siirtymiseen kaupungista 3 kaupunkiin 0 ja seitsemäntenä päivänä hän vierailee 10 nähtävyydessä kaupungissa 0. Jian-Jian vierailemien nähtävyyksien yhteismäärä on $20 + 30 + 10 = 60$, mikä on suurin mahdollinen määrä nähtävyyksiä, joissa Jian-Jia voi vierailla 7 päivässä aloittaen kaupungista 2.

kaupunki	nähtävyyksien määrä
0	10
1	2
2	20
3	30
4	1

päivä	tapahtuma
1	vierailu kaupungin 2 nähtävyyksissä
2	liikkuminen kaupungista 2 kaupunkiin 3
3	vierailu kaupungin 3 nähtävyyksissä
4	liikkuminen kaupungista 3 kaupunkiin 2
5	liikkuminen kaupungista 2 kaupunkiin 1
6	liikkuminen kaupungista 1 kaupunkiin 0

päivä	tapahtuma
7	vierailu kaupungin 0 nähtävyyksissä

Tehtävä

Toteuta funktio `findMaxAttraction`, joka laskee suurimman mahdollisen nähtävyyksien määrän, joissa Jian-Jia voi vierailla.

- `findMaxAttraction(n, start, d, attraction)`
 - `n`: kaupunkien lukumäärä.
 - `start`: aloituskaupungin indeksi.
 - `d`: päivien lukumäärä.
 - `attraction`: taulukko kokoa `n`; `attraction[i]` on kaupungin `i` nähtävyyksien määrä, missä $0 \leq i \leq n - 1$.
 - Funktion tulee palauttaa suurin mahdollinen määrä nähtävyyksiä, joissa Jian-Jia voi vierailla.

Osatehtävät

Kaikissa osatehtävissä $0 \leq d \leq 2n + \lfloor n/2 \rfloor$, ja jokaisen kaupungin nähtävyyksien määrä on epänegatiivinen.

Lisärajoitukset:

osatehtävä	pisteet	n	nähtävyyksien määrä kaupungissa (t)	aloituskaupunki
1	7	$2 \leq n \leq 20$	$0 \leq t \leq 1\,000\,000\,000$	ei rajoitusta
2	23	$2 \leq n \leq 100\,000$	$0 \leq t \leq 100$	kaupunki 0
3	17	$2 \leq n \leq 3\,000$	$0 \leq t \leq 1\,000\,000\,000$	ei rajoitusta
4	53	$2 \leq n \leq 100\,000$	$0 \leq t \leq 1\,000\,000\,000$	ei rajoitusta

Toteutus

Sinun täytyy palauttaa tarkalleen yksi tiedosto nimeltä `holiday.c`, `holiday.cpp` tai `holiday.pas`. Tämän tiedoston tulee toteuttaa yllä kuvattu funktio käyttäen seuraavia runkoja. Sinun täytyy myös ottaa mukaan otsikkotiedosto `holiday.h` C/C++-toteutuksessa.

Huomaa, että tulos voi olla suuri ja funktion `findMaxAttraction` palautusarvo on 64-bittinen kokonaisluku.

C/C++-ohjelma

```
long long int findMaxAttraction(int n, int start, int d,
```

```
int attraction[]);
```

Pascal-ohjelma

```
function findMaxAttraction(n, start, d : longint;  
attraction : array of longint): int64;
```

Esimerkkitarkistin

Esimerkkitarkistin lukee syötteen seuraavassa muodossa:

- rivi 1: n, start, d.
- rivi 2: attraction[0], ..., attraction[n-1].

Esimerkkitarkistin tulostaa funktion `findMaxAttraction` palautusarvon.