Atostogos

Jian-Jia kitais metais nori atostogauti Taivane. Atostogaudamas jis planuoja keliauti iš vieno miesto į kitą ir juose apeiti įdomias vietas.

Taivane yra n miestų, visi jie išsidėstę palei vieną greitkelį. Miestai sunumeruoti nuosekliai nuo 0 iki $n-1$. Miestui i ($0 < i < n-1$) gretimų miestų numeriai yra $i-1$ ir $i+1$. Vienintelis miestas, gretimas miestui 0, yra miestas 1, o vienintelis miestas, gretimas miestui $n-1$ yra miestas $n-2$.

Kiekviename mieste yra tam tikras skaičius įdomių vietų. Jian-Jia turi d dienų atostogų ir per atostogas nori aplankyti kiek galima daugiau įdomių vietų. Jis jau nusprendė, kuriame mieste pradės savo atsotogas. Kiekvieną atostogų dieną Jian-Jia gali arba pervažiuoti į gretimą miestą, arba aplankyti visas tame mieste esančias įdomias vietas. Tą pačią dieną jis negali ir pervažiuoti į kitą miestą ir lankytis įdomiose vietose. Jian-jia *neis į jau aplankytą įdomią vietą antrą kartą*, net jei tą patį miestą lankys kelis kartus.

Padėkite Jian-Jia taip suplanuoti savo atostogas, kad per jas aplankytų kuo daugiau įdomių vietų.

Pavyzdys

Tarkime, Jian-Jia turi 7 atostogų dienas, o Taivane yra 5 miestai (išvardinti lentelėje žemiau). Atostogas jis pradeda 2 mieste. Pirmąją dieną Jian-Jia aplankys 20 tame mieste esančių įžymių vietų. Antrąją dieną Jian-Jia pervažiuos iš 2 į 3 miestą. Trečiąją dieną aplankys 30 tame mieste esančių įdomių vietų. Tris tolesnes dienas jis keliaus iš 3 į 0 miestą. Septintąją dieną aplankys 10 tame mieste esančių įdomių vietų.

Per savo atostogas Jian-Jia aplankys 20 + 30 + 10 = 60 įdomių vietų. Tai yra maksimalus galimas skaičius, jei jis pradeda atostogas 2 mieste.

miestas	įdomių vietų skaičius
0	10
1	2
2	20
3	30
4	1

diena	veiksmas
1	aplankomos 2 miesto įdomios vietos
2	vykstama iš 2 miesto į 3
3	aplankomos 3 miesto įdomios vietos
4	vykstama iš 3 miesto į 2
5	vykstama iš 2 miesto į 1

diena	veiksmas
6	vykstama iš 1 miesto į 0
7	aplankomos 0 miesto įdomios vietos

Užduotis

Parašykite funkciją `findMaxAttraction`, kuri suskaičiuotų, kiek daugiausiai įdomių vietų galės aplankyti Jian-Jia.

- `findMaxAttraction(n, start, d, attraction)`
  - `n`: miestų skaičius.
  - `start`: pradinio miesto numeris.
  - `d`: dienų skaičius.
  - `attraction`: n ilgio masyvas; `attraction[i]` reiškia mieste i esančių įžymių vietų skaičių; galioja $0 \leq i \leq n-1$.
  - Funkcija turi grąžinti maksimalų įdomių vietų, kurias gali aplankyti Jian-Jia skaičių.

Dalinės užduotys

Visoms dalinėms užduotims galioja $0 \leq d \leq 2n + \lfloor n/2 \rfloor$. Įdomių vietų skaičius kiekviename mieste yra neneigiamas.

Papildpomi ribojimai:

dalinė užduotis	taškai	n	didžiausias įdomių vietų skaičius mieste (t)	pradinis miestas
1	7	$2 \leq n \leq 20$	$0 \leq t \leq 1\,000\,000\,000$	nėra ribojimų
2	23	$2 \leq n \leq 100\,000$	$0 \leq t \leq 100$	miestas 0
3	17	$2 \leq n \leq 3\,000$	$0 \leq t \leq 1\,000\,000\,000$	nėra ribojimų
4	53	$2 \leq n \leq 100\,000$	$0 \leq t \leq 1\,000\,000\,000$	nėra ribojimų

Reikalavimai

Pateikite failą, pavadintą `holiday.c`, `holiday.cpp` arba `holiday.pas`. Faile turi būti aukščiau aprašyta funkcija. Įterpkite antraštinį failą `holiday.h`, jei programuojate C/C++.

Atkreipkite dėmesį, kad grąžinamas rezultatas gali būti didelis, todėl `findMaxAttraction` grąžina 64 bitų sveikąjį skaičių.

Programuojantiems C/C++

```
long long int findMaxAttraction(int n, int start, int d,
int attraction[]);
```

Programuojantiems Paskaliu

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

Pavyzdinis vertintojas

Pavyzdinis vertintojas skaito duomenis tokiu formatu:

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

Pavyzdinis vertintojas išveda `findMaxAttraction` grąžinamą reikšmę.