

D. Garden Decorations | Sodo dekoracijos

Užduoties pavadinimas	Sodo dekoracijos
Laiko apribojimas	7 sekundės
Atminties apribojimas	1 gigabaitas

Kiekvieną dieną eidama į mokyklą ir iš jos, Deitė praeina gatvę su N namų, kurie yra sunumeruoti nuo 0 iki $N - 1$. Šiuo metu name i gyvena žmogus i . Kad būtų įdomiau, gyventojai nusprendė pasikeisti namais tarpusavyje. Žmogus, kuris persikraustys į namą i , yra žmogus a_i (ir šis žmogus šiuo metu gyvena name a_i).

Prie kiekvieno namo stovi vienoda paukščio statula. Statulai turi dvi įmanomas būsenas: arba sparnai yra *išskleisti* (tarsi paukštis skristų), arba *suskleisti* (tarsi paukštis stovėtų ant žemės).

Gyventojai turi stiprius įsitikinimus apie tai, kaip jų statula turėtų atrodyti, ir atsisako persikraustyti į naują namą, kol statula naujajame name neatrodys taip pat kaip ir senajame name. Deitė nori padėti gyventojams sutvarkyti jų statulas, kad jie galėtų persikraustyti.

Šiam tikslui įgyvendinti, Deitė elgiasi taip: kai eina šia gatve (pavyzdžiui, eidama į mokyklą ar iš jos), Deitė pažiūri į statulas, kurias praeina, ir pakoreguoja kai kurias iš jų (išskleisdama arba suskleisdama paukščių sparnus). Kadangi Deitė yra labai užimta tiek mokykloje, tiek namuose, **ji neprisimena paukščių būsenų, kurias matė praėjusiais kartais eidama šia gatve.**

Laimei, Deitė užsirašė sąrašą a_0, a_1, \dots, a_{N-1} , taigi, ji žino, kuris gyventojas kur persikrausto.

Padėkite Deitei sugalvoti strategiją, kuri nurodys, kurias statulas pakeisti, kad jos atitiktų gyventojų norus. Deitė gatve gali pereiti daugiausia 60 kartų, bet kad surinktumėte maksimalų taškų skaičių, ji turėtų gatve pereiti mažiau kartų.

Įgyvendinimas

Tai yra kelių paleidimų (angl. multirun) užduotis, tai reiškia, kad jūsų programa bus paleista kelis kartus.

Kiekvieno paleidimo metu, pirmiausia perskaitykite eilutę su dviem sveikaisiais skaičiais w ir N , tai yra pasivaikščiojimo numeris ir namų skaičius. Pirmo programos paleidimo metu $w = 0$, antro paleidimo metu $w = 1$ ir taip toliau (išsamiau paaiškinta toliau užduotyje).

Antroje įvesties eilutėje yra N sveikųjų skaičių a_0, a_1, \dots, a_{N-1} , kurie reiškia, kad žmogus, kuris persikraustys į namą i , šiuo metu gyvena name a_i .

a_i indeksai suformuoja seką (angl. permutation), t.y. kiekvienas skaičius nuo 0 iki $N - 1$ sekoje pasirodo lygiai vieną kartą. Atkreipkite dėmesį, kad gyventojas gali nuspręsti nesikraustyti, t.y., $a_i = i$ yra leidžiama.

Gyventojai persikrausto tik vieną kartą. Tai reiškia, kad fiksuotam testo atvejui, N reikšmė ir a_i reikšmių sąrašas nesikeis per visus jūsų programos paleidimus.

Pirmasis programos paleidimas

Pirmojo programos paleidimo metu $w = 0$. Šio paleidimo metu, jūs turėtumėte išvesti vieną sveikąjį skaičių W ($0 \leq W \leq 60$), reiškiantį, kiek kartų Deitė turėtų pereiti per gatvę. Tada programa turėtų pasibaigti. Toliau programa bus paleista dar W kartų.

Tolimesni programos paleidimai

Kitame programos paleidime, $w = 1$, dar kitame $w = 2$ ir taip toliau, o paskutinio programos paleidimo metu $w = W$.

Kai perskaitysite w , N ir a_0, a_1, \dots, a_{N-1} , Deitė pradės eiti gatve.

- Jeigu w yra nelyginis skaičius, Deitė eis iš namų į mokyklą ir praeis pro namus tvarka $0, 1, \dots, N - 1$.

Jūsų programa dabar turėtų perskaityti eilutę su b_0 reikšme, kuri gali būti 0 (sparnai suskleisti) or 1 (sparnai išskleisti), šis skaičius nurodo dabartinę statulos būseną prie namo 0. Kai perskaitysite b_0 , tuomet turėtumėte išvesti eilutę su 0 arba 1, t.y. naująja b_0 verte.

Tuomet jūsų programa turėtų perskaityti eilutę su b_1 , tai reprezentuoja 1 statulos būseną, ir išvesti naująją b_1 reikšmę. Toks procesas tęsiasi visiems N namų. Kai paskutiniam namui atliekami visi veiksmai (t. y. perskaitoma ir išvedama b_{N-1} reikšmė), **jūsų programa turėtų pasibaigti.**

Atkreipkite dėmesį, kad jūsų programa gali perskaityti naują reikšmę b_{i+1} tik po to, kai išvedėte reikšmę b_i .

- Jeigu w yra lyginis skaičius, Deitė eis iš mokylos namo ir praeis pro namus atvirkščia tvarka $N - 1, N - 2, \dots, 0$.

Tai reiškia, kad pradėdama skaityti ir išvesti nuo b_{N-1} , tada b_{N-2} ir taip toliau iki b_0 .

Kai $w = 1$, įvesties reikšmės b_0, b_1, \dots, b_{N-1} yra pradinės statulų būsenos (ir taip pat norimos kiekvieno namo gyventojų statulų būsenos). Kai $w > 1$, įvesties reikšmės b_0, b_1, \dots, b_{N-1} bus tokios

reikšmės, kurios buvo nustatytos jūsų programos praėjusio programos paleidimo metu.

Galiausiai, po paskutinio programos paleidimo, b_i reikšmė turi būti lygi pradinei b_{a_i} reikšmei visiems i , kitu atveju bus gautas atsakymas „Neteisingas atsakymas“ (angl. „Wrong Answer“).

Detalės

Jeigu programos visų $W + 1$ paleidimų laikų *suma* viršija laiko limitą, jūsų programos pateikimas bus vertinamas kaip „Klaida – viršytas laiko limitas“ (Time Limit Exceeded).

Po užklauskos pateikimo standartinei išvesčiai būtinai atlikite procedūrą flush, kitaip jūsų programa gali būti įvertinta kaip „Klaida – viršytas laiko limitas“ (Time Limit Exceeded). Python kalboje `input()` atlieka flush procedūrą automatiškai. C++ kalboje `cout << endl;` taip pat atlieka flush procedūrą (ir, žinoma, išspausdina naują eilutę); jei naudojate `printf`, naudokite `fflush(stdout)`.

Apribojimai ir vertinimas

- $2 \leq N \leq 500$.
- Galima naudoti ne daugiau kaip $W \leq 60$ raundų.

Jūsų sprendimas bus testuojamas su keliomis testų grupėmis, kurių kiekviena verta tam tikro taškų skaičiaus. Kiekviena testų grupė yra sudaryta iš testų rinkinio. Norėdami gauti taškus už testų grupę, turite išspręsti visus tos grupės testų atvejus.

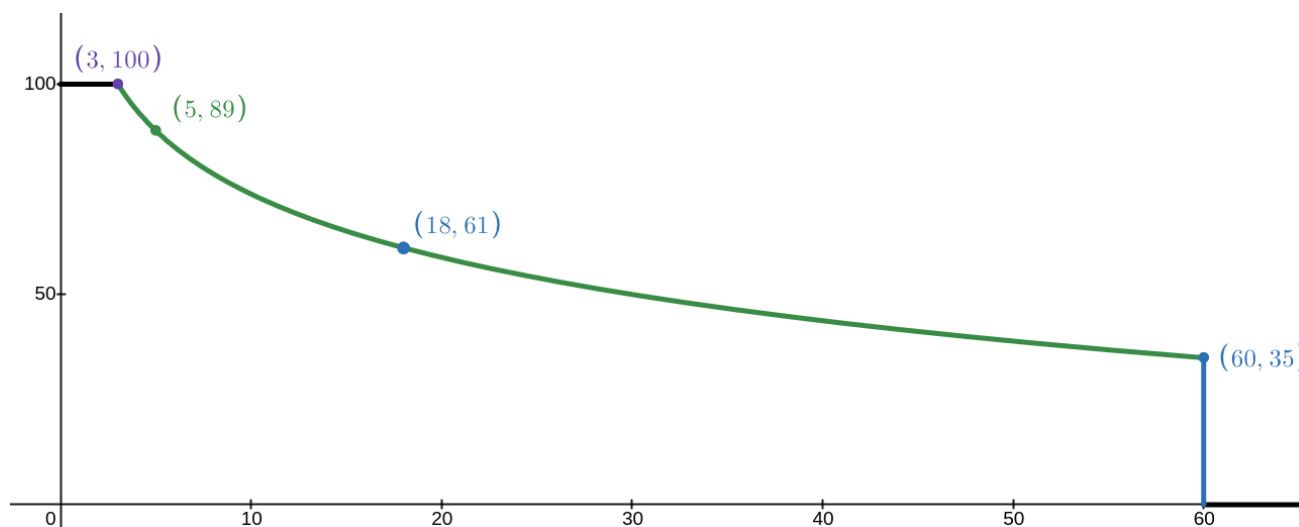
Grupė	Taškai	Apribojimai	Komentaras
1	10	$N = 2$	-
2	24	$N \leq 15$	-
3	9	$a_i = N - 1 - i$	-
4	13	$a_i = (i + 1) \bmod N$	operatorius mod reiškia dalybos iš N liekaną
5	13	$a_i = (i - 1) \bmod N$	operatorius mod reiškia dalybos iš N liekaną
6	31	Jokių papildomų apribojimų	-

Kiekvienai testų grupei, kurią programa išspręs teisingai, jums bus skiriami taškai pagal žemiau pateiktą formulę:

$$\text{score} = S_g \cdot \left(1 - \frac{1}{2} \log_{10}(\max(W_g, 3)/3)\right),$$

čia S_g yra maksimalus taškų įvertis konkrečiai testų grupei, o W_g yra maksimali W vertė, naudota bet kuriam testavimo variantui testų grupėje. Jūsų įvertis kiekvienai testų grupei bus suapvalintas iki artimiausio sveiką skaičiaus.

Žemiau pateiktame grafike matoma taškų, gaunamų už užduotį, priklausomybė nuo W . Jūsų programa gaus tokį įvertinimą, jei išspręs visas testų grupes su tokia pačia W reikšme. Tiksliau, kad surinktumėte 100 taškų šioje užduotyje, jums reikia išspręsti kiekvieną testą su $W \leq 3$.



Testavimo įrankis

Kad būtų lengviau išbandyti sprendimą, pateikiame paprastą įrankį, kurį galite atsisiųsti. Žr. „Attachments“ Kattis užduoties puslapio apačioje. Įrankiu naudotis neprivaloma. Atkreipkite dėmesį, kad oficiali vertinimo programa Kattis skiriasi nuo testavimo įrankio

Kad naudotumėte įrankį, sukurkite įvesties failą, tarkime, „sample1.in“, kuris turėtų prasidėti skaičiumi N , toliau turėtų sekti eilė N skaičių, nurodančių seką, ir kita eilutė su N bitų (0 arba 1), nurodančių pradines paukščių būsenas. Pavyzdžiui:

```
6
1 2 0 4 3 5
1 1 0 0 1 0
```

Python programai, tarkime `solution.py` (paprastai paleidžiamai su `pypy3 solution.py`):

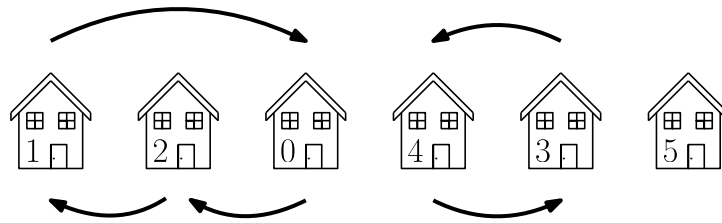
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ programai, pirmiausia ją sukompiliuokite (pavyzdžiui, su `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) ir tada paleiskite:

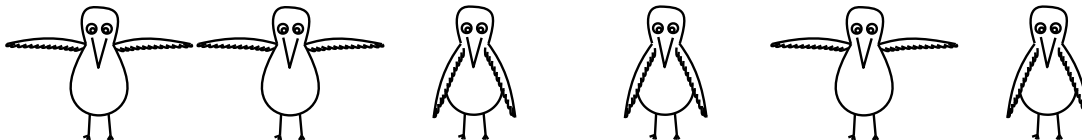
```
python3 testing_tool.py ./solution.out < sample1.in
```

Pavyzdys

Pavyzdyje turime toliau pateiktą žmonių namuose seką:

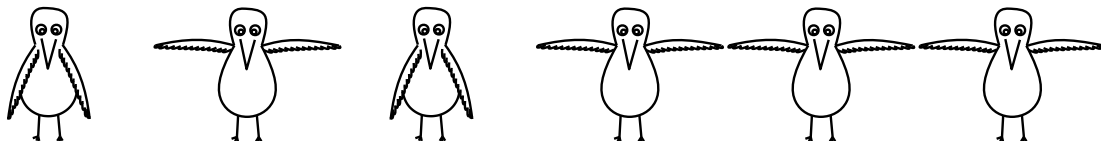


Pirmą kartą, kai pavyzdinė programa paleidžiama (su $w = 0$), ji išveda $W = 2$, kas reiškia, kad Deitė eis gatve du kartus (o programa bus paleista dar du kartus). Prieš pirmą pasivaikščiojimą, paukščiai namų soduose atrodo taip:



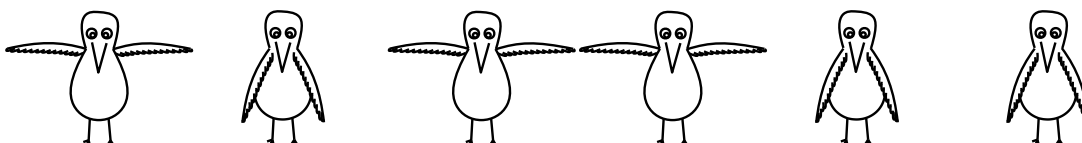
Tuomet programa paleidžiama su $w = 1$: tai reiškia pirmą Deitės pasivaikščiojimą. Ji praeina paukščius po vieną, pradėdama nuo kairės, ir gali pakeisti jų būsenas. Pavyzdinė programa turi išvesti i -tojo paukščio būseną prieš perskaitant $(i + 1)$ -tojo paukščio būseną.

Po Deitės atvykimo į mokyklą, paukščių būsenos atrodo taip:



Paskutinis programos paleidimas (su $w = 2$) atrodo taip: Deitė grįžta atgal iš mokyklos namo. Prisiminkite, kad šiuo atveju Deitė praeis pro paukščius iš dešinės į kairę ir atliks veiksmus su jais priešinga tvarka! Tai reiškia, kad Deitė turi nustatyti i -tojo paukščio būseną prieš perskaitant $(i - 1)$ -tojo paukščio būseną.

Kai Deitė pabaigia savo pasivaikščiojimą, paukščiai atrodo taip:



Iš tiesų, tai yra teisinga konfigūracija. Pavyzdžiui, paukščio statula nr. 3 (t.y. ketvirtoji iš kairės) yra išskleistais sparnais (dabar $b_3 = 1$), kas yra teisinga, nes žmogus nr. 4 persikraustys čia ($a_3 = 4$) ir šis žmogus pradiniu laiko momentu turėjo paukštį išskleistais sparnais (pradiniu laiko momentu $b_4 = 1$).

Vertinimo programos (grader) išvestis	Jūsų išvestis
0 6	
1 2 0 4 3 5	
	2

Vertinimo programos (grader) išvestis	Jūsų išvestis
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

Vertinimo programos (grader) išvestis	Jūsų išvestis
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1