

Liftas

Laiko limitas: 10 sekundžių

Atminties limitas: 64 MiB (vienam procesui)

Tai - komunikacijos uždavinys.

Yra pastatas, kurio aukštai sunumeruoti nuo 0 iki N . Aukštas 0 yra apatinis aukštas, o virš aukšto N yra stogas. Taigi, įskaitant stogą, pastatas turi $N + 2$ aukštus.

Kiekviename aukšte f ($0 \leq f \leq N$) gyvena lygiai vienas gyventojas, ir tas aukštas turi slaptą sveikąjį skaičių v_f ($0 \leq v_f \leq 3$), kurį žino tik jo gyventojas. Ant šio pastato stogo gyvena Karlsonas. Tikslas, kad Karlsonas nustatytų kiek įmanoma daugiau reikšmių $v_0, v_1, \dots, v_N$, o kiekvienas gyventojas **bendradarbiauja**, kad būtų kuo daugiau reikšmių, kurias jis gali atkurti.

Norėdami bendrauti, pastato gyventojai naudojami liftu tokiu būdu:

Liftas pradeda kelionę aukšte 0 ir kyla tik į viršų. Lifo viduje yra mygtukai, pažymėti aukštais nuo 1 iki N . Iš pradžių nė vienas mygtukas nėra paspaustas. Kai mygtukas paspaudžiamas, jis lieka paspaustas visam laikui. Liftas sustoja ir atsidaro f aukšte, jei $f = 0$, arba f aukštą atitinkantis mygtukas buvo anksčiau paspaustas. Aplankęs aukščiausią aukštą, kurio mygtukas buvo paspaustas (arba 0 aukštą, jei nė vienas mygtukas niekada nebuvo paspaustas), liftas važiuoja tiesiai ant stogo, nesustodamas jokiuose likusiuose aukštuose.

Kai liftas sustoja aukšte f :

1. Gyventojas mato visus jau paspaustus mygtukus.
2. Remdamasis tik šia informacija ir reikšme v_f , jis gali paspausti bet kuriuos dar nepaspaustus mygtukus, kurie yra **griežtai aukščiau** už jo paties aukštą.
3. Liftas pakyla į pirmą aukštesnį aukštą, kurio atitinkamas mygtukas buvo paspaustas (arba ant stogą, jei visi aukštai, atitinkantys paspaustus mygtukus, jau buvo aplankyti).

Jei liftas nesustoja kuriame nors aukšte, jo gyventojas negali paspausti jokių mygtukų.

Kai liftas pasiekia stogą, Karlsonas mato tik galutinį paspaustų mygtukų rinkinį. Naudodamasis šia informacija, jis bando atkurti kuo daugiau reikšmių $v_0, v_1, \dots, v_N$.

Reikšmės v yra fiksuojamos prieš pradedant vykdyti jūsų programą ir proceso metu nesikeičia.

Implementacijos detalės

Bus T testų.

Pateikite vieną failą, kuriame realizuosite šias dvi funkcijas.

Pirmoji funkcija, kurią turite realizuoti, yra:

```
std::vector<int> press_buttons(int subtask, int N,
                               int f, int v, std::vector<int> p)
```

- $subtask$: dalinės užduoties numeris, kuriai ši funkcija yra iškviečiama, kur $0 \leq subtask \leq 4$;
- N : paskutinio aukšto numeris;
- f : dabartinis aukštas, kur $0 \leq f \leq N$;
- v : reikšmė v_f , užrašyta aukšte f ;
- p : šiuo metu paspausti mygtukai, didėjimo tvarka.

Ši funkcija iškviečiama, kai liftas atsidaro aukšte f (jei $f = 0$, arba aukštą f atitinkantis mygtukas buvo paspaustas ankstesnėje stotelėje). Ši funkcija nebus iškviesta aukštui, jei jo atitinkamas mygtukas nebuvo paspaustas anksčiau.

Grąžinama reikšmė yra naujų mygtukų, kuriuos reikia paspausti, masyvas. Reikšmių tvarka grąžinamame masyve nėra svarbi. Kiekvienas paspaustas mygtukas x turi tenkinti šias sąlygas:

- $f < x \leq N$;
- x dar nėra patekęs į p ir x patenka lygiai vieną kartą grąžinamame masyve.

Antra funkcija, kurią turite realizuoti:

```
std::vector<int> answer(int subtask, int N, std::vector<int> p)
```

- $subtask$: dalinės užduoties numeris, kuriam ši funkcija yra iškviečiama, kur $0 \leq subtask \leq 4$;
- N : paskutinio aukšto numeris;
- p : galutiniai paspausti mygtukai didėjimo tvarka.

Ši funkcija iškviečiama lygiai vieną kartą testui, liftui pasiekus stogą.

Grąžinamas masyvas turi būti $N + 1$ ilgio, jo i -tasis elementas turi būti lygus v_i , o reikšmės, kurių Karlsonas **negali atkurti**, turi būti pakeistos į -1 .

Svarbu: Žmonės pastate negali bendrauti jokių kitu būdu, kaip tik per paspaudžiamus mygtukus. Kad tai būtų užtikrinta, jūsų programa bus vykdoma kaip $N + 2$ atskiri procesai, po vieną kiekvienam aukštui ir vienas stogui. Kiekvieno aukšto procese funkcija `press_buttons` bus iškviesta daugiausiai T kartų, o stogo procese funkcija `answer` – lygiai T kartų. Todėl jūsų

programa neturėtų remtis jokia bendrai naudojama būseną (pvz., globaliais kintamaisiais). Kiekvienam procesui yra prieinama 64 MiB atminties, kuri yra atskira nuo kitų procesų atminties. Visi iškvietimai (iki $(N + 1) \times T_{\text{press_buttons}}$ ir lygiai T_{answer}) turi būti įvykdyti per 10 sekundžių.

Ribojimai

- $N = 60$
- $T \leq 10\,000$
- $0 \leq v_i \leq 3$ kiekvienam $0 \leq i \leq N$

Pavyzdys

Pavyzdys turi vieną testą su šiomis v reikšmėmis:

```
1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 1
1 0 1 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 0 1 1
```

Bendravimo pavyzdys:

Dalyvio programa	Komisijos programa
	<code>press_buttons(0, 60, 0, 1, {})</code>
<code>return {2}</code>	
	<code>press_buttons(0, 60, 2, 0, {2})</code>
<code>return {13, 42}</code>	
	<code>press_buttons(0, 60, 13, 0, {2, 13, 42})</code>
<code>return {}</code>	
	<code>press_buttons(0, 60, 42, 1, {2, 13, 42})</code>
<code>return {}</code>	
	<code>answer(0, 60, {2, 13, 42})</code>
<code>return {1, -1, 0, -1, -1, ..., -1}</code>	

Šis bendravimas teisingai atkuria 0-ąją ir 2-ąją reikšmes. Visos kitos grąžintos reikšmės yra lygios -1 , nes Karlsonas jų neatkūrė.

Dalinės užduotys

Dalinė užduotis	Taškai	Papildomi ribojimai	Aukštų skaičius, kurį reikia atkurti, kad gautumėte visus taškus
0	0	Pavyzdys.	–
1	15	$0 \leq v_i \leq 1$ $v_0 = v_N = 1$ Jei $v_i = 0$, tada $v_{i+1} = 1$ kiekvienam $0 \leq i < N$	60
2	35	$0 \leq v_i \leq 1$	40
3	15	$0 \leq v_i \leq 2$	30
4	35	$0 \leq v_i \leq 3$	25

Vertinimas

Jeigu jūsų programa grąžina neteisingai atkurtą reikšmę arba atlieka negalimą operaciją (pvz., grąžina neteisingo ilgio `vector`) bet kuriame užduoties dalies testo atvejuje, jūsų pateiktis už tą užduoties dalį gaus nulį taškų.

Teste *atkurtų reikšmių skaičius* yra pozicijų, kurių grąžinta reikšmė nėra -1 , kiekis. Tegul K yra mažiausias atkurtų reikšmių kiekis tarp visų užduoties dalies testų (kiekviena dalinė užduotis turi tik 1 didelį testą, kuriame yra iki 10 000 testų). Taškai už kiekvieną dalinę užduotį apskaičiuojami pagal žemiau pateiktą lentelę.

Dalinė užduotis	K dydis	Taškai
0	-	0
1	$K < 60$	$0.25K$
	$K \geq 60$	15
2	$K < 30$	$0.5K$
	$30 \leq K < 40$	$2K - 45$
	$K \geq 40$	35
3	$K < 30$	$0.5K$
	$K \geq 30$	15
4	$K < 20$	$0.7K$
	$20 \leq K < 25$	$4K - 66$
	$K \geq 25$	35

Pavyzdinis vertintojas

Pavyzdinis vertintojas (ang. grader) atlieka visus funkcijų iškvietimus visiems testams vieno vykdymo metu.

Įvesties formatas:

- eilutė 1: trys sveikieji skaičiai – testų skaičius T , paskutinio aukšto numeris N ir dalinės užduoties numeris S ;
- eilutė $1 + i$: $N + 1$ sveikieji skaičiai $v_0, v_1, \dots, v_N$, kur v_f yra reikšmė, užrašyta aukšte f .

Išvesties formatas:

- eilutė i : vienas sveikasis skaičius – teisingai atkurtų v reikšmių skaičius testo atvejuje i ;
- eilutė $T + 1$: galutiniai taškai.

Išsamesnei informacijai pakeiskite kintamąjį `DETAILED` iš `false` į `true` pirmoje vertintojo eilutėje.

Galite leisti vertintojui sugeneruoti testinius atvejus (reikšmes v_f) už jus, pakeisdami kintamąjį `AUTO_GENERATE` iš `false` į `true` antroje vertintojo eilutėje.