

Užduotis: ZER

Nuliai ir vienetai



BOI 2025, Diena 0. Leidžiamas atminties kiekis: 256 MB.

2025.04.25

Turime dvejetainę seką a_0, \dots, a_{n-1} . Sekos nariai nėra žinomi, bet jūs galite paklausti bet kurių dviejų skirtingų jos narių sumos. Jūsų užduotis – atkurti dvejetainę seką pateikiant ne per daug užklausų.

Užklausos

Ši užduotis yra interaktyvi. Jūs turite parašyti programą, kuri rastų sprendimą siųsdama užklausas ir gaudama atsakymus į jas per standartinį įvesties ir išvesties įrenginį.

Visų pirma jūsų programa iš standartinio įvesties įrenginio turi nuskaityti dvejetainės sekos ilgį n ($3 \leq n \leq 200\,000$).

Toliau jūsų programa turi užduoti klausimus tokiu formatu:

- Norėdama sužinoti sekos elementų a_i ir a_j sumą, jūsų programa į standartinį išvesties įrenginį turi išvesti:

$i \ j$

, kur i ir j yra skirtingi sveikieji skaičiai ($0 \leq i \neq j \leq n-1$).

- Atsakymas bus pateiktas taip:

s

, kur s yra sveikasis skaičius ($0 \leq s \leq 2$) lygus $a_i + a_j$.

Kai jau žinote seką, į standartinį išvesties įrenginį turite išvesti dvi eilutes tokiu formatu:

n

$a_0 \ a_1 \ \dots \ a_{n-1}$

, kur n nurodo sekos ilgį, o a_0, a_1, \dots, a_{n-1} – jos elementus. Tada programa turi baigti darbą. Jei programa užduos papildomų klausimų, sprendimas gali būti laikomas klaidingu (gauti klaidą: **Wrong answer**).

Neužmirškite, kad po kiekvienos užklausos reikia priverstinai išvalyti standartinio išvesties įrenginio buferį C++ naudojant `cout.flush()` (arba `fflush(stdout)`, jei naudojate `printf`) arba pridėti `flush = True` parametrą į Python `print` komandą. Kitu atveju jūsų programa gali viršyti leistiną vykdymo laiką (gauti klaidą `Time Limit Exceeded`).

Programa negali skaityti jokių failų ar kitų operacinės sistemos resursų. Testavimo tikslais jūs galite papildomą informaciją išvesti į standartinį klaidų įrenginį (standard error stream), bet neužmirškite, kad rašymas į jį užtrunka papildomai.

C++ įvestis/išvestis naudojant `iostream`

Programos antraštėje reikia pridėti (`#include <iostream>`). Kiekvienos išvesties eilutės pabaigoje reikia pridėti `std::endl`. Pradinės komunikacijos pavyzdys:

```
std::cin >> n;
std::cout << i << ' ' << j << std::endl;
std::cin >> sum;
```

C++, įvestis/išvestis naudojant `stdio`

Programos antraštėje reikia pridėti (`#include <stdio.h>`). Išvedę eilutę į ekraną turite iškviešti `fflush(stdout)`. Pradinės komunikacijos pavyzdys:

```
scanf("%d", &n);
printf("%d %d\n", i, j);
fflush(stdout);
scanf("%d", &sum);
```

Python

Visuose `print` funkcijos kvietiniuose reikia pridėti parametą `flush = True`. Pradinės komunikacijos pavyzdys:

```
n = int(input())
print(f"{i} {j}", flush = True)
sum = int(input())
```

Interaktyvaus sprendimo pavyzdys

Veiksmas	Parametrai	Išvestis	Paaškinimas
nuskaityti n	–	5	$n = 5$
suma	0 1	1	$a_0 + a_1 = 1$
suma	1 2	1	$a_1 + a_2 = 1$
suma	3 4	2	$a_3 + a_4 = 2$, todėl $a_3 = a_4 = 1$
suma	0 3	2	$a_0 + a_3 = 2$, todėl $a_0 = 1$, ir todėl $a_1 = 0$ bei $a_2 = 1$
atsakymas	5 1 0 1 1 1		Teisingas atsakymas, $m = 4 \leq n = 5$ užklausų, 100% taškų.

Vertinimas

Dalinė užduotis	Ribojimai	Taškai
1	$3 \leq n \leq 1000$	50
2	$3 \leq n \leq 200\,000$	50

Dvejtainė seka **neprivalo** būti fiksuota programos vykdymo pradžioje. Ieškoma seka gali būti pakeista programos vykdymo metu, bet garantuojama, kad atsakymai į visus užduotus klausimus visada galios.

Jei sprendama testą jūsų programa pateiks m užklausų, tada jūs gausite tiek procentų taškų numatytų už testą:

Užklausa	Taškų procentinė dalis
$m \leq n$	100% taškų už testą
$m = n + 1$	80% taškų už testą
$n + 1 < m \leq n^2 - n$	50% taškų už testą
$m > n^2 - n$	0% taškų už testą (Wrong answer)