

Uzdevums: ZER

Nulles un vieninieki



BOI 2025, diena 0. Atmiņas ierobežojums: 256 MB.

2025.04.25

Eksistē bināra virkne a_0, \dots, a_{n-1} . Jūs nezināt tās elementus, bet varat vaicāt summu jebkuriem diviem atšķirīgiem elementiem. Jūsu uzdevums ir atrast šo virkni, izmantojot ne pārāk daudz vaicājumu.

Komunikācija

Šis ir interaktīvs uzdevums. Jums jāuzraksta programma, kas atrod korektu šī uzdevuma risinājumu un komunicē ar žūrijas programmu (**interactor**), lasot no standarta ievades un rakstot uz standarta izvadi.

Komunikācijas sākumā jūsu programmai jānolasa vesels skaitlis n ($3 \leq n \leq 200\,000$) no standarta ievades, kas apzīmē virknes garumu.

Tad jūsu programmai jājautā vaicājumi šādi:

- Jūsu programmai jāizvada viena rinda formā

$i \ j$

uz standarta izvadi, kur i un j ir dažādi veseli skaitļi ($0 \leq i \neq j \leq n-1$). Šī rinda attēlo vienu vaicājumu par elementiem a_i un a_j .

- Atbilde tiks dota šādi:

s

kur s ir vesels skaitlis ($0 \leq s \leq 2$) vienāds ar $a_i + a_j$.

Kad jūs noskaidrosiet virkni, izvadiet divas rindas formā

n

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

uz standarta izvadi, kur n ir virknes garums un a_0, a_1, \dots, a_{n-1} ir tās elementi. **Pēc tam programmai darbība ir jābeidz. Jautājot vēl vaicājumus, var tikt iegūts Wrong answer vērtējums.**

Paturiet prātā, ka pēc katra vaicājuma vai atbildes jums ir jāsinchronizē (*flush*) izvades *buferi* izmantojot `cout.flush()` (vai `fflush(stdout)`, ja izmantojat `printf`) C++ valodā vai jāpievieno `flush = True` argumentu `print` komandai Python valodā. Citādi jūsu programma var saņemt `Time Limit Exceeded` vērtējumu.

Jūsu programma nedrīkst vērt valā jebkādas failus vai izmantot citus resursus. Tā var izmantot standarta kļūdas izvadi atklādošanas nolūkiem, bet jāņem vērā, ka rakstīšana uz šo plūsmu patērē laiku.

C++, ievade/izvade, izmantojot plūsmas

Jums jāiekļauj atbilstošā galvene (`#include <iostream>`). Katra izvades rinda jāpabeidz ar `std::endl`. Skatiet sekojošo piemēru ar komunikācijas sākumu.

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

C++, ievade/izvade, izmantojot stdio

Jums jāiekļauj atbilstošā galvene (`#include <stdio.h>`). Pēc katra izvades rindas jāizsauc `fflush(stdout)`. Skatiet sekojošo piemēru ar komunikācijas sākumu.

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

Python

Katrā `print` funkcijas izsaukumā jābūt `flush = True` argumentam. Skatiet sekojošo piemēru ar komunikācijas sākumu.

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

Komunikācijas piemērs

Darbība	Parametri	Izvade	Apraksts
lasīt n	–	5	$n = 5$
summa	0 1	1	$a_0 + a_1 = 1$
summa	1 2	1	$a_1 + a_2 = 1$
summa	3 4	2	$a_3 + a_4 = 2$, tātad $a_3 = a_4 = 1$
summa	0 3	2	$a_0 + a_3 = 2$, tātad $a_0 = 1$, un tālāk $a_1 = 0$ un $a_2 = 1$
atbilde	5 1 0 1 1 1		Pareiza atbilde, $m = 4 \leq n = 5$ jautājumi, 100% punkti.

Vērtēšana

Apakšuzdevums	Ierobežojumi	Punkti
1	$3 \leq n \leq 1000$	50
2	$3 \leq n \leq 200\,000$	50

Žūrijas programmai (`interactor`) nav jānofiksē ievades virkne a_0, \dots, a_{n-1} komunikācijas sākumā. Tā var mainīt virknes elementus, ja tas nav pretrunā ar tiktāl uz vaicājumiem atgrieztajām atbildēm. Citiem vārdiem, žūrijas programma ir adaptīva.

Apzīmējot ar m jūsu programmas maksimālo vaicājumu skaitu, jūs saņemsiet sekojošu procentu no punktiem par konkrētu testu:

Vaicājumi	Procenti no punktiem
$m \leq n$	100% punkti par testu
$m = n + 1$	80% punkti par testu
$n + 1 < m \leq n^2 - n$	50% punkti par testu
$m > n^2 - n$	0% punkti par testu (Kļūda)