

Ülesanne: ZER

Nullid ja ühed



BOI 2025, Päev 0. Mälulimiit: 256 MB.

2025.04.25

On binaarne jada a_0, \dots, a_{n-1} . Sa ei tea selle elemente, aga saad küsida kahe erineva elemendi summat. Sinu ülesandeks on taastada kogu jada kasutades mitte liiga palju päringuid.

Suhtlus

See on interaktiivne ülesanne. Sinu ülesandeks on kirjutada programm, mis leiab sobiva jada ja suhtleb hindamisprogrammiga, lugedes standardsisendist ja kirjutades standardväljundisse.

Alguses peab programm lugema standardsisendist täisarvu n ($3 \leq n \leq 200\,000$), see on jada pikkus.

Edasi peab programm esitama päringuid järgneval moel:

- Programm peab standardväljundisse väljastama ühe rea kujul

$i \ j$

kus i ja j on erinevad täisarvud ($0 \leq i \neq j \leq n-1$). See rida tähistab ühte päringut elementide a_i ja a_j kohta.

- Vastus antakse kujul:

s

kus s on täisarv ($0 \leq s \leq 2$), mis on võrdne avaldisega $a_i + a_j$.

Kui oled jada taastanud, siis väljasta standardväljundisse kaks rida kujul

n

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

kus n on jada pikkus ja a_0, a_1, \dots, a_{n-1} on selle elemendid. **Seejärel peaksid lõpetama oma programmi töö. Lisapäringute esitamine võib lõppeda Wrong answer tulemusega.**

Pange tähele, et pärast igat päringut või vastust tuleb väljundipuhver tühjendada, kasutades käsklust `cout.flush()` (või `fflush(stdout)` kui kasutada `printf-i`) keeles C++, või Pythonis lisades `print`-käsklusele argumendi `flush = True`. Muidu võib lahendus saada tulemuseks Time Limit Exceeded (“ajalimiit ületatud”).

Programm ei tohi avada ühtegi faili ega kasutada muid ressursse. Programm tohib silumise otstarbel kasutada veaväljundit (“standard error”), kuid pange tähele, et sinna kirjutamine võtab ka aega.

C++, sisend/väljund vooge kasutades

Lisa vastav päis (`#include <iostream>`). Iga rea väljastamise lõpus kutsu välja `std::endl`. Vaata järgnevat näidet interaktsiooni kohta.

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

C++, sisend/väljund kasutades stdio

Lisa vastav päis (`#include <stdio.h>`). Iga rea väljastamise lõpus kutsu välja `fflush(stdout)`. Vaata järgnevat näidet interaktsiooni kohta.

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

Python

Igal print funktsiooni väljakutsel peaks olema `flush = True` argumendina. Vaata järgnevat näidet interaktsiooni kohta.

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

Näidisinteraktsioon

Tegevus	Parameetrid	Väljund	Kirjeldus
n lugemine	–	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	$a_3 + a_4 = 2$, seega $a_3 = a_4 = 1$
summa	0 3	2	$a_0 + a_3 = 2$, seega $a_0 = 1$, ja $a_1 = 0$ and $a_2 = 1$
vastus	5 1 0 1 1 1		Õige vastus, $m = 4 \leq n = 5$ päringut, 100% punktidest.

Hindamine

Alamülesanne	Piirangud	Punkte
1	$3 \leq n \leq 1000$	50
2	$3 \leq n \leq 200\,000$	50

Hindamisprogramm **ei pea** kogu sisendjada a_0, \dots, a_{n-1} kogu interaktsiooni alguses fikseerima. Hindamisprogramm võib muuta käesoleva jada elemente, seni kuni see on kooskõlas kõigi seni esitatud päringute vastustega. Teisisõnu on hindamisprogramm adaptiivne.

Olgu m maksimaalne arv päringuid, mida sinu programm teeb, sa saad järgmise protsendi antud testi punktide arvust:

Päringuid	Protsent punktidest
$m \leq n$	100% punktidest testi eest
$m = n + 1$	80% punktidest testi eest
$n + 1 < m \leq n^2 - n$	50% punktidest testi eest
$m > n^2 - n$	0% punktidest testi eest (Wrong answer)