ispitni centar

PRAVA MJERA ZNANJA

DRŽAVNO TAKMIČENJE 2021.

ŠIFRA UČENIKA

SREDNJA ŠKOLA

PROGRAMIRANJE

UKUPAN BROJ OSVOJENIH BODOVA

Test pregledala/pregledao

.....

.....

Podgorica, ... 20..... godine

Uputstva takmičarima

Ovo takmičenje sastoji se od rješavanja **3 problemska zadatka** u vremenu od **4 sata** (240 minuta). Zadatke je potrebno rješavati u jednom od sljedećih programskih jezika: Pascal, C, C++ ili Java. Takmičari koji koriste Pascal moraju programirati u programskom alatu **FreePascal ili TurboPascal.** Takmičari u C-u i C++-u moraju koristiti programske alate **CodeBlocks ili GCC**. Za programski jezik Java predviđena je upotreba platforme **Eclipse**. Dozvoljeno je koristiti editor po izboru i pomoću navedenih alata prevoditi izvorni kod u izvršnu datoteku.

Tokom takmičenja **ne smijete komunicirati** ni sa jednom osobom, osim dežurne osobe takmičenja. To znači da morate **raditi samostalno** i **ne smijete koristiti Internet**. Takođe, zabranjena je upotreba bilo kakvih ranije napisanih programa ili dijelova programa.

Po isteku vremena predviđenog za takmičenje, na desktopu u folderu sa imenom ***Takmicenje2021*** moraju se nalaziti datoteke sa snimljenim izvornim kôdovima rješenja. Nakon takmičenja, komisija će testirati vaša rješenja na ranije izabranim test podacima i dodijeliti vam određeni broj bodova. Na kraju svakog zadatka dati su primjeri test podataka. Ti primjeri služe da bi vam tekst zadataka bio što je moguće jasniji te za provjeru formata ulaza i izlaza, a ne služe za provjeru ispravnosti vašeg programa. Ako vaš program radi na tim primjerima, to **nije garancija** da će raditi na službenim podacima za testiranje.

Zadaci ne nose jednak broj bodova. Lakše i brže rješivi zadaci nose manje bodova, dok teži nose više bodova. Ukupan broj bodova na nekom zadatku jednak je zbiru bodova test podataka koji se poklapaju sa službenim rješenjem. Ukupan broj bodova jednak je zbiru bodova na svim zadacima.

Sve informacije o zadacima (ime zadatka, vremensko i memorijsko ograničenje, način bodovanja) možete naći na uvodnoj stranici sa naslovom *Zadaci*. Ako vam nije jasno nešto u vezi načina organizacije ovog takmičenja, odmah postavite pitanje dežurnom da vam to razjasni.

Tokom cijelog takmičenja možete postavljati pitanja dežurnom u vezi zadataka ili koristiti sistem za pojašnjenja (clarifications). Dozvoljena su pitanja **koja razjašnjavaju nejasnoće u tekstu zadatka**. Ne smijete postavljati pitanja u vezi rješavanja zadataka. Prije nego postavite pitanje, pročitajte još jednom zadatak, jer je moguće da ste u prethodnom čitanju preskočili dio teksta zadatka.

Nepoštovanje ovih pravila ili nepridržavanje formata izlaznih podataka rezultiraće nepovratnim gubitkom bodova. Nemojte štampati ništa što se u zadatku ne traži, kao npr. poruke tipa 'Rjesenje je:' ili 'Unesite brojeve' i slično!

Srećno i uspješno takmičenje!

Zadaci

Zadatak	Zadatak1	Zadatak2	Zadatak3
Izvorni kôd	Zadatak1.java zadatak1.pas zadatak1.c zadatak1.cpp	Zadatak2.java zadatak2.pas zadatak2.c zadatak2.cpp	Zadatak3.java zadatak3.pas zadatak3.c zadatak3.cpp
Memorijsko ograničenje	256 MB	512 MB	128 MB
Vremensko ograničenje (po test podatku)	2 sekunde	1 sekunda	2 sekunde
Broj test podataka	32	16	26
Broj bodova (po test podatku)	1	2	1 ili 2 (16x1+10x2)
Ukupno bodova	**32**	**32**	**36**

Zadatak 1 – Planine

Kompanija ProgKom treba da postavi radne stanice po planinskim vrhovima planete Pithon. Planine su označene brojevima od 1 to n i poređane su u jednom redu slijeva udesno, pri čemu je visina planine i jednaka H_i. ProgKom je odlučio da postavi 3 radne stanice na planinama x, y i z, tako da na planini y bude osnovna stanica a na planinama x i z budu pomoćne stanice. Planina x mora biti lijevo od planine y, dok planina z mora biti desno od planine y. Obje planine x i z moraju biti strogo niže od planine y. ProgKom je poznata po svojoj preciznosti, pa vas je angažovala da prije početka radova odredite koliko ima mogućnosti za postavljanje radnih stanica.

Ulazni podaci

U prvom redu ulaza nalazi se n – broj planina ($1 \le n \le 3 \cdot 10^5$). U drugom redu nalazi se n cijelih brojeva H_i ($0 \le H_i \le 10^{18}$, $1 \le i \le n$) – visine planina.

Izlazni podaci

U jedinom redu izlaza štampati jedan cio broj – koliko ima trojki planina na koje je moguće postaviti radne stanice.

Bodovanje:

- Podzadatak 1 (2 boda) – H je neopadajući niz (tj. $H_i \le H_j$ za $i \le j$)
- Podzadatak 2 (2 boda) – $0 \le H_j \le 1$
- Podzadatak 3 (4 boda) – $0 \le H_j \le 99$
- Podzadatak 4 (7 bodova) – $n \le 500$
- Podzadatak 5 (7 bodova) – $n \le 10^4$
- Podzadatak 6 (6 bodova) – $0 \le H_j \le 10^5$
- Podzadatak 7 (4 boda) – nema dodatnih ograničenja

Test primjeri

Ulaz	Izlaz	Objašnjenje
5 0 1 1 0 1	2	Ovaj primjer je validan za sve podzadatke osim za podzadatak 1. Tražene trojke su (1, 2, 3) i (1, 3, 4).
6 500 20 900 0 900 700	7	Ovaj primjer je validan samo za podzadatke 4, 5, 6 i 7. Tražene trojke su (1, 3, 4), (1, 3, 6), (1, 5, 6), (2, 3, 4), (2, 3, 6), (2, 5, 6) i (4, 5, 6).

Rješenje:

Podzadatak 1

Kako je niž H neopadajućim ne mogu postojati planine y i z takve da je $y < z$ i $H_y > H_z$, pa je odgovor 0.

Podzadatak 2

Sve visine su 0 ili 1, pa mora biti $H_x = H_z = 0$ i $H_y = 1$. Neka sL označava broj nula u H[1..i-1] a sR broj jedinica u H[i+1..n] . Sačuvajmo ove vrijednosti u posebnom nizu za O(n). Ako je glavna stanica na planini y za koju je $H_y=1$, tada imamo sL kandidata za x i sR kandidata za z, pa je broj izbora za dato y jednak sL x sR. Sumirajmo sve moguće izbore za y. Složenost: O(n)

Podzadatak 3:

Moguće visine za glavnu stanicu su planine y za koje je $0 \le h \le 99$. Stvarana visina planine nije bitna, već samo da li je visna veća ili manja of Hy. Posmatrajmo sve Hi koji je Hi < h kao 0 a sve za koje Hi ≥ h kao 1. Tada smo sveli zadatak na Podzadatak 1. Složenost: $O(n \cdot Max(X_i))$

Podzadatak 4:

Iteracija kroz sve moguće trojke (x,y,z) i provjera da li zadovoljavaju uslove. Složenost: $O(n^3)$.

Podzadatak 5:

Direktna primjena ideje iz podzadatka 3 ne prolazi ako je $Max(Hi) = 10^{18}$. Međutim, provjeravamo samo h = Hi za neke $1 \le i \le n$, pa ima najviše n različitih vrijednosti, tako da modifikovana verzija rješenja za podzadatak 3 prolazi. Složenost: $O(n^2)$.

Podzadatak 6:

U prethodnim podzadacima smo izračunavali sL i sR za svaku visinu . Možemo iskoristiti neku od struktura za „upite opsega" (range queries) kao npr, drvo Fenwicka ili segmentno drvo. Neka je query(h) broj visina koje su, do tog trenutka, najviše h i neka je update(h) dodavanje visne h. Iteracijom prolazimo kroz niz H i za svaki indeks i izračunavamo sL = $query(H_i - 1)$. Zatim dodamo trenutnu visinu są $update(H_i)$. Na sličan način računamo i sR. Odgovor je suma proizvoda sL × sR. Složenost: $O(n \cdot \log(\max(H_i)))$.

Podzadatak 7:

Direktna primjena ideje iz podzadatka 6 ne prolazi ako je $Max(H_i) = 10^{18}$. Međutim, možemo diskretizovati vrijednosti H_i tj. najmanji H_i je 1, sljedeći je 2, itd. i primjeniti ideju iz podzadatka 6-

Zadatak 2 – Trouglovi

Maša je na času matematike upravo učila o krugu opisanom oko trougla. Kao i uvijek, Maša sve posmatra na svoj način, pa je upitala nastavnicu Anu: “Da li se može reći da je trougao upisan u krug?”. Nastavnica je odgovorila potvrdno, a zatim zadala Maši sljedeći zadatak: na krugu obima L dato je n različitih tačaka. Koliko ima upisanih trouglova koji sadrže centar kruga, pri čemu centar može biti i na stranci trougla, a tjemena su im u datim tačkama?
Pomozite Maši i napišite program koji će pronaći traženi broj trouglova.

Ulazni podaci

U prvom redu ulaza nalaze se 2 cijela broja n i L ($3 \le n \le 300\,000$, $n \le L \le 10^9$). Izaberimo proizvoljnu tačku S na krugu. Tada se svaka tačka A na krugu može predstaviti brojem x koji predstavlja rastojanje, po krugu, u smjeru kretanja kazaljke sata, od S do A, $0 \le x < L$. Ovaj broj nazivamo koordinatom tačke A. Drugi red ulaza sadrži n različitih cijelih brojeva koji predstavljaju koordinate $x_1, x_2, ..., x_n$ ($0 \le x_i < L$).

Izlazni podaci

Štampati jedan cio broj – broj upisanih trouglova koji sadrže centar kruga. Centar se može naći i na stranici trougla.

Test primjeri

Ulaz	Izlaz
3 10 0 2 1	0
10 10 0 1 2 3 4 5 6 7 8 9	60

Rješenje:

Trougao ABC ne sadrži centar ako postoji prečnik koji ne presijeca trougao. Odredićemo koliko ima trouglova koji ne sadrže centar i oduzeti ga od ukupnog broja trouglova. Kod takvog trougla, rastojanje između dvije susjedne tačke po krugu mora biti bar L/2, pri čemu trougao može sadržati najviše jedan takav par.

Sortiramo tačke tako da je $x_1 < x_2 < \ldots < x_n$. Imamo dva slučaja:

- par (x_i, x_j), gdje je $i < j$: tada za treće tjeme imamo mogućih $i - 1 + n - j$ tačaka (tj. tačke $x_1, \ldots, x_{i-1}$ i $x_{j+1}, \ldots, x_n$)
- par (x_i, x_j), gdje je $i > j$: tada za treće tjeme imamo mogućih $i - j - 1$ (tj. tačke $x_{j+1}, \ldots, x_{i-1}$).

  Skup odgovarajućih j možemo implementirati pomoću dva brojača („two pointers”), a sumiranje po njima je suma aritmetičkog niza. Složenost je $O(n \cdot \log n)$ za sortiranje + $O(n)$ za dva brojača, što u zbiru daje $O(n \cdot \log n)$.

Zadatak 3 – Berza

Na planeti ProgramBet različite berze nude razne koeficijente za iste ishode takozvanih binarnih opcija. Binarna opcija ima samo dva moguća ishoda: raste ili opada. Za one koje nisu upoznati sa pravilima berze, koeficijent x na neki ishod znači da ćete ako uplatite 1 euro na taj ishod, od berze dobiti tačno x eura; ako ste pogriješili u procjeni ishoda događaja, nećete osvojiti ništa.
Na ovoj planeti moguće je trgovati binarnu opciju tako da vam je profit zagarantovan. Planeta ProgramBet je poznato galaktičko finasijsko tržište i ima n berzi koje nude razne koeficijente za binarnu opciju. Prigodno, berze su označene brojevima od 1 do n. Koeficijent koji i-ta berza nudi za rast označavamo sa p_i, a koeficijent koji ta ista berza nudi za opadanje označavamo sa q_i. Pravila na ovoj planeti su neuobičajena: možete izabrati bilo koji podskup ponuđenih koeficijenata, pa čak možete uzeti i oba ishoda koje nudi ista berza. Ograničeni ste pravilom da vaš ulog na svaki ishod mora biti tačno 1 euro i da nije moguće napraviti više kupovina na neki ishod u istoj berzi. U slučaju rasta, osvojićete p_i eura od svake berze i u kojoj ste izabrali rast. Slično, u slučaju opadanja, osvojićete q_i eura iz svake berze i u kojoj ste izabrali opadanje. U oba slučaja već ste uložili 1 euro.
Koji je najveći zagarantovani profit, nezavisno od ishoda, ako ste optimalno birali koeficijente?

Ulazni podaci

U prvom redu ulaza nalazi se n – broj berzi na ProgramBetu ($1 \le n \le 100000$). Svaki od sljedećih n redova sadrži koeficijente koje nude berze, tj. u redu i+1 nalaze se realni brojevi p_i i q_i koji redom označavaju koeficijent rasta, odnosno opadanja binarne opcije koji nudi berza i ($1.0 \le p_i, q_i \le 1000.0$). Koeficijenti neće sadržati više od 4 cifre nakon decimalne tačke.

Izlazni podaci

U jedinom redu štampati najveći mogući zagarantovani profit zaokružen na tačno 4 decimalna mjesta. Štampanje na tačno 4 decimalna mjesta zavisi od programskog jezika koji koristite:

- C i C++: `printf("%.4f",x);`
- C++:
```
#include <iomanip>
…
cout << fixed << setprecision(4) << x;
```
- Java: `System.out.printf("%.4f",x);`
- Pascal: `writeln(x:0:4);`
- Python 3: `print("%.4f"%x);`

Bodovanje:

- Podzadatak 1 (6 bodova) – $n \le 10$
- Podzadatak 2 (10 bodova) – $n \le 1\,000$
- Podzadatak 3 (20 bodova) – nema dodatnih ograničenja

Test primjeri

Ulaz	Izlaz	Objašnjenje
4 1.4 3.7 1.2 2 1.6 1.4 1.9 1.5	0.5000	Optimalni izbor je da u berzi 1 biramo opadanje i da u berzama 3 i 4 biramo rast. Ako bude rast, zaradićemo 1.6+1.9-3=0.5 a ako bude opadanje zaradićemo 3.7-3=0.7, pa je zagarantovani dobitak 0.5.

Rješenje:
Potrebno je naći maksimum izraza ($\Sigma p_i - n_a - n_b$, $\Sigma q_j - n_a - n_b$), gdje je p_i iznos predviđen za rast opcije, q_i za opadanje opcije, n_a i n_b redom predstavljaju broj izabranih kvota za rast odnosno opadanje opcije.

Podzadatak 1
Isprobati sve moguće podskupove, jer je $n \leq 10$.

Podzadatak 2
Ako oduzmemo 1 od svake kvote, izraz postaje ($\Sigma p_i - n_b$, $\Sigma q_j - n_a$). Za fiksirano n_a i n_b tačan izabrani skup za rast opcije ne utiče na optimalan izbor za opadanje. Sortirajmo vrijednosti p_i i q_i, pa za svaki par n_a i n_b izaberemo n_a najvećih p_i i n_b najvećih q_i. Složenost je $O(n^2)$

Podzadatak 3
Posmatrajmo dati izraz za fiksiranu vrijednost n_a i za različite vrijednosti n_b (tj. kao funkciju od n_b). Povećavanjem n_b izraz se uvećava sve dok $\Sigma q_j - n_b$ ne postane veća od $\Sigma p_i - n_b$, kada počne da opada. Možemo koristiti ternarno traženje da bi odredili maksimum. Ili, može se koristiti i binarno traženje po nizu razlika dobijenim u uzastopnim tačkama n_b. Izračunati prefiksne sume da bi za $O(1)$ odredili traženu virijednost za date n_a i n_b. Složenost je $O(n \cdot \log n)$.