

 ispitni centar
PRAVA
MJERA
ZNANJA

DRŽAVNO TAKMIČENJE 2023.

SREDNJA ŠKOLA

PROGRAMIRANJE

Autorka/autor testa

Recenzentkinja/recenzent

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**. Takmičari u C-u i C++-u moraju koristiti programski alat **CodeBlocks**. 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 **Takmicenje2023** moraju se nalaziti datoteke sa snimljenim izvornim kôdovima rješenja. Vaša rješenja možete upload-ovati na sistem za takmičenje, ako je dostupan, i provjeriti koliko bodova ste dobili. Nakon takmičenja, komisija će testirati vaša rješenja još jednom. Na kraju svakog zadatka dati su primjeri test podataka. Ti primjeri služe da bi vam tekst zadataka bio što je moguće jasniji i 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.

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 s 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. 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!

VAŽNO za jezik Java!

Ne kreirajte pakete za vaše zadatke, već koristite podrazumijevani (default) paket.

Zadaci

Zadatak	Zadatak1 (A)	Zadatak2 (B)	Zadatak3 (C)
Izvorni kôd	A.java A.pas A.c A.cpp	B.java B.pas B.c B.cpp	C.java C.pas C.c C.cpp
Memorijsko ograničenje	256MB	256MB	256MB
Vremensko ograničenje (po test podatku)	3 sekunde	1 sekunda	1 sekunda
Ukupno bodova	33	33	34

Zadatak 1



U domu boravi N stanara. Kao dio zajedničkog druženja oni gledaju televiziju. Na raspolaganju imaju M programa označenih brojevima od 1 do M . Svaki od penzionera ima omiljeni i omraženi TV program.

Ako se trenutno na televizoru prikazuje omraženi program nekog stanara, on će ustati, vrlo sporo odšetati do televizora i prebaciti na svoj omiljeni program. Nakon toga će se vratiti u svoju udobnu fotelju. Ako postoji više stanara kojima je trenutni program omražen, ustaće najmlađi od njih, a ostali će ostati da sjede.

Naravno, nakon jedne promjene programa, može se naći drugi stanar kojem se novi program ne sviđa pa će i on promijeniti program. Budući da su stanari tvrdoglavi, ovo se može nastaviti u nedogled.

Za zadane omiljene i omražene programe stanara i početni program koji se prikazuje na TV-u, odredite broj promjena programa nakon kojih će svi stanari zadovoljno sjediti.

Ulaz:

U prvom redu standardnog ulaza nalaze se tri prirodna broja N , M i P ($1 \leq N$, $M \leq 100000$, $1 \leq P \leq M$), broj stanara, broj TV programa i početni program na televizoru.

U svakom od idućih N redova nalaze se dva prirodna broja a_i i b_i ($1 \leq a_i$, $b_i \leq M$, $a_i \neq b_i$), omiljeni i omraženi program svakog stanara. Stanari su u ulazu poredani od najmlađeg prema najstarijem.

Izlaz:

U jedini redak ispišite traženi broj promjena programa. Ako će se promjene nastaviti u nedogled, ispišite -1.

Primjer:

Ulaz	Izlaz
3 3 1 1 2 2 3 3 1	-1
4 5 2 1 3 2 3 3 2 5 1	3

Podzadaci:

U test podacima ukupno vrijednima 50% bodova važiće $1 \leq N$, $M \leq 1000$.

Rješenje:

Uočimo da problem možemo predstaviti grafom: čvorovi će biti televizijski programi, a parovi omiljenog i omraženog programa svakog stanara usmjere grane od omraženog prema omiljenom programu.

U slučaju da je neki program omražen za više stanara (pa bi tada trebali i imati više grana koji izlaze iz njega svaki prema svojem omiljenom programu) dovoljno je pamtiti samo jednu granu: onaj najmlađeg stanara jer on ustaje i mijenja program u tom slučaju.

Budući da je sada odabir mijenjanja programa jednoznačno određen, lako je doći do rješenja: krećemo se po konstruiranom grafu krenuvši od zadanog programa P i brojimo koliko smo trenutno programa promijenili (označimo taj broj sa K) i koje smo programe već gledali (prošli ih našim kretanjem u grafu).

Ako dođemo na program na kojem smo već bili, očito je da će se opet isti programi ponavljati (kažemo da smo ušli u ciklus) pa ispisujemo -1.

Ako dođemo do programa koji nije omražen nikome (iz njega ne izlazi grana prema nekom drugom programu), programi se više neće mijenjati pa je ukupni broj promjena programa K (kretanja u grafu) naše rješenje.

Zadatak 2



Dat je niz od N prirodnih brojeva. Pronađite podniz uzastopnih elemenata dužine barem K koji ima najveći mogući prosjek.

Napomena: prosjek podniza je zbir svih brojeva u podnizu podijeljen s njegovom dužinom.

Ulaz:

Prva linija ulaza sadrži dva prirodna broja N ($1 \leq N \leq 300000$) i K ($1 \leq K \leq N$).

U drugom retku nalazi se N prirodnih brojeva a_i ($1 \leq a_i \leq 100000$).

Izlaz:

U prvi i jedini red ispišite najveći mogući prosjek. Dozvoljeno apsolutno odstupanje od službenog rješenja je ± 0.001 .

Primjer:

Ulaz	Izlaz
4 1 1 2 3 4	4.000000
4 2 2 4 3 4	3.666666
6 3 7 1 2 1 3 6	3.333333

Podzadaci:

U 50% test podataka N neće biti veći od 5 000.

Rješenje

Direktno odgovaranje na pitanje koji je najveći mogući prosjek je težak problem. Umjesto toga pokušaćemo da odgovoriti na pitanje je li najveći mogući prosjek veći ili jednak nekoj vrijednosti P .

Ako to uspijemo onda zadatak možemo riješiti binarnim pretraživanjem po vrijednosti P . I taj problem izgleda teško, no primjetimo ovo: ako svim brojevima u nizu oduzmemo X , tada će se i prosjek svih podnizova smanjiti za tačno X . Sada možemo svim brojevima u nizu oduzeti vrijednost P i ono što nas zanima jest postoji li podniz s prosjekom većim ili jednakim 0. Taj problem je jednostavniji jer je to ekvivalentno pitanju postoji li podniz sa sumom većom ili jednakom nuli.

Ako napravimo niz prefiksni suma S , tada je suma nekog podniza $[a, b]$ jednaka $S[b] - S[a-1]$. Interesuje nas da li postoji podniz dužine barem K , takav da je zbir njegovih elemenata veći od 0.

Ako napravimo još jedan pomoćni niz M , takav da $M[i]$ čuva $\min\{S[0], S[1], \dots, S[i]\}$ tada podniz uzastopnih elmenata sa najvećom sumom čija je dužina veća od K i koji se završava na poziciji b , možemo da dobijemo kao $S[b] - M[b-K-1]$.

Ukupna složenost algoritma je $O(n \log(n))$.

Zadatak 3



Marko je našao novi posao u rudniku. Njegov zadatak je da kontrolira mašine koje obrađuju rudu. U rudniku postoji k mašina. Svaka mašina može da obrađuje nekoliko vrsta rude. Pošto je posao prilično automatizovan možemo pretpostaviti da svaka od mašina može da obradi proizvoljno mnogo rude svakog dana. Marko želi da se pokaže kao dobar radnik i da obradi što više rude prvog dana. Kako bi to postigao, treba da odabere tri vrste mašina koje će da koristi.

Do mašina jedan za drugim, dolaze vagoni sa rudom koja se zatim obrađuju u mašinama redoslijedom kojim dolaze. Ako vagon sadrži vrstu rude koju mašine koje je Marko odabrao ne mogu da obrade, on će biti nezadovoljan i prestaće da radi.

Pomozite Marku da odabere tri mašine pomoću kojih će obraditi najveći broj vagona. Unaprijed je poznato za svaki od vagona koje vrste rude sadrži, kao i za svaku od mašina koje vrste rude može da obradi.

Ulaz:

Prva linija ulaza sadrži tri cijela broja $N(1 \leq N \leq 20000)$, $K(1 \leq K \leq 1000)$ i $S(1 \leq S \leq 1000)$ koji predstavljaju broj vagona, broj različitih tipova rude kao i broj mašina koje postoje u rudniku. Rude su numerisane brojevima od 1 do K , a mašine brojevima od 1 do S . Narednih S linija sadrže opise tipova rude koju mašine mogu da obrade. U liniji $i+1$ nalazi se opis ruda koje može da obradi i -ta mašina. Opis se sastoji od niza brojeva koji opisuju rude koje mašina može da obradi. Opis mašine se završava kad se učitava 0. Posljednja linija ulaza sadrži N prirodnih brojeva koji predstavljaju vrstu rude koje sadrži svaki od vagona redoslijedom kojim dolaze na obradu. Za svaki tip rude postoji najviše 10 vrsta mašine koje mogu da obrade tu vrstu rude.

Izlaz:

Prva linija izlaza sadrži jedan broj koji predstavlja maksimalan broj vagona koji mogu da se obrade pomoću tri mašine. Druga linija sadrži 3 broja koji predstavljaju mašine koje Marko treba da izabere. Ako postoji više mogućih izbora štampati bilo koji od njih.

Primjer:

Ulaz:	Izlaz:
13 5 4	11
1 0	2 1 4
4 5 0	
5 3 0	
2 5 0	
4 5 2 5 5 4 1 1 5 4 5 3 3	

Podzadaci:

Za 50% bodova važiće $N \leq 10000$ i $S \leq 300$.

Rješenje

Pošto nije bitno koja od mašina koje je Marko odabrao treba da obradi rudu, kao prvu mašinu može da uzme neku od mašina koja može da obradi tip rude koji se nalazi u prvom vagonu. Takvih mašina je najviše 10. Tu mašinu koristi sve dok ne dođe do vrste rude koju ne može da obradi prvom mašinom. Tada odabere drugu vrstu mašine i nastavi da obrađuje rudu sve dok ne dođe do vrste rude koju ne može da obradi ni prvom ni drugom mašinom. Tada izabere treću mašinu i nastavi sa obradom rude. Potrebno je ispitati kombinacije koje se dobijaju za svaki mogući izbor mašina primjenom prethodnih postupka. Pošto za svaku vrstu rude postoji najviše 10 mašina koje mogu da obrade tu vrstu rude postoji najviše 10^3 takvih kombinacija.

Neophodno je izvršiti preprocesiranje kako bi bilo moguće efikasno pronaći sve mašine koje mogu da obrade neku vrstu rude. Kako bi se to postiglo moguće je napraviti niz vektora tako da se u vektor na poziciji i smjeste sve mašine koje mogu da obrade rudu i .

Ukupna složenost algoritma je $O(1000n) = O(n)$.