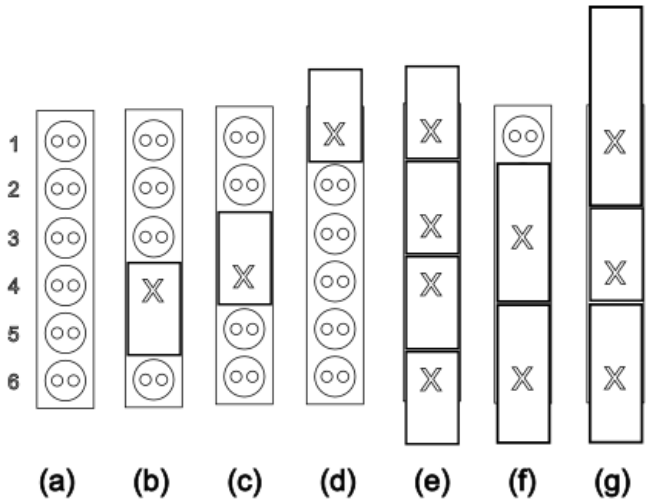
LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
VECĀKĀ (11. - 12. KLAŠU) GRUPA

Latvijas informātikas olimpiāde

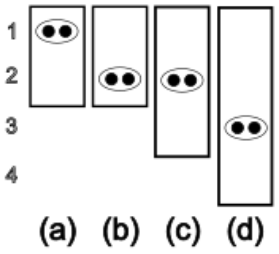
Adapteru rinda

Mūsdienu elektroniskajām iekārtām ir nepieciešama regulāra uzlāde, kas parasti tiek veikta, izmantojot dažādu izmēru adapterus. Šajā uzdevumā interesēsimies par adapteru ieslēgšanu kontaktligzdu blokā, kur kontaktligzdas izvietotas vienā rindā (1. attēlā (a) parādīts sešu kontaktligzdu bloks). Katra adaptera kontaktdakšiņa ir novietota tā, ka, ieslēdzot to kontaktligzdā, adaptera garākā mala būs paralēla kontaktligzdu bloka garākajai malai, turklāt kontaktligzdā adapteru var ieslēgt divos variantos – viens variants no otra atšķiras ar pagriešanu par 180 grādiem. Pieņemsim, ka katra adaptera garākās malas garums ir kontaktligzdu augstuma daudzkārtnis.

Piemēram, ja adapters (2. att. (a) un (b)) ir divas vienības augsts, tad to var ieslēgt iepriekš aplūkotā kontaktligzdu bloka ceturtajā kontaktligzdā divos variantos – pēc ieslēgšanas tajā bez tās, kurā tiek iesprausta kontaktdakšiņa (attēlos apzīmēta ar „x"), tiks aizsegta attiecīgi piektā (1. att. (b)) vai trešā (1. att. (c)) kontaktligzda. Labā ziņa ir tā, ka adaptera korpuss var iet arī ārpus kontaktligzdu bloka (1. att. (d)), aizsedzot pēc iespējas mazāk vai nemaz neaizsedzot liekas kontaktligzdas. Ņemot to vērā, šajā piemērā sešu kontaktligzdu blokā varētu ieslēgt, augstākais, četrus šāda izmēra adapterus (1. att. (e)).



1. attēls: Kontaktligzdu bloks un adapteru ieslēgšana blokā



2. attēls: Adapteru izmēri

Katru adapteru raksturo tā garums l_i un kontaktdakšiņas atrašanās pozīcija adaptera korpusā p_i ($1 \leq p_i \leq l_i$). 2. attēlā (attiecīgi, (a), (b), (c), (d)) parādīti adapteri (2;1), (2;2), (3;2) un (4;3).

Saprotams, ka kontaktligzdu blokā vienlaicīgi ieslēgto adapteru skaits ir atkarīgs no adapteru izmēriem. Ja visi adapteri ir trīs vienības gari un kontaktdakšiņa atrodas otrajā (vidējā) pozīcijā (2. att. (c)), tad sešu kontaktligzdu blokā varēs ieslēgt ne vairāk kā divus šādus adapterus (piemēram, ieslēdzot tos $3.$ un $6.$ kontaktligzdā, 1. att. (f)).

Uzrakstiet datorprogrammu, kas dotam kontaktligzdu skaitam blokā un dotajam adapteru komplektam nosaka, cik no tiem vienlaikus var ieslēgt kontaktligzdu blokā un kā tieši tie jāieslēdz!

Ievaddati

Pirmajā rindā doti divi naturāli skaitļi – kontaktligzdu skaits blokā $N (N \leq 10^9)$ un adapteru skaits $A (A \leq 2 \cdot 10^5)$.

Nākamajās A rindās katrā dots viena adaptera apraksts – divi naturāli skaitļi l_i (adaptera garums, $1 \leq l_i \leq 10^9$) un kontaktdakšiņas atrašanās pozīcija adaptera korpusā p_i ($1 \leq p_i \leq l_i$)).

Starp katriem diviem blakus skaitļiem ievaddatos ir tukšumzīme.

Izvaddati

Izvaddatu pirmajā rindā jābūt naturālam skaitlim M – lielākajam adapteru, kurus no dotajiem var vienlaikus ieslēgt dotajā kontaktligzdu blokā, skaitam.

Nākamajās A rindās katrā jābūt veselam skaitlim robežās no $-N$ līdz N. Katram $i (1 \leq i \leq A)$ izvaddatu $i+1$-ajā rindā jābūt raksturotam, kā kontaktligzdu blokā jāieslēdz adapters, kurš ievaddatos dots kā i-tais pēc kārtas.

Šim skaitlim jābūt:

- 0, ja attiecīgais adapters **nav ieslēgts** kontaktligzdu blokā;
- $v (1 \leq v \leq N)$, ja attiecīgais adapters ir ieslēgts kontaktligzdā v, orientējot adapteru **tā, kā dots ievaddatos**;
- $-v (1 \leq v \leq N)$, ja attiecīgais adapters ir ieslēgts kontaktligzdā v, orientējot adapteru **pagrieztu par 180 grādiem** attiecībā pret to, kā dots ievaddatos.

Nenulles skaitļu kopskaitam jābūt M un blokā ieslēgto adapteru konfigurācijai jābūt neprerunīgai. Ja iespējamas vairākas derīgas adapteru konfigurācijas ar lielāko M vērtību, izvadiet informāciju par jebkuru no tām.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Adapteri**

Piemēri

Ievaddati	*Izvaddati*	*Piezīme*
6 5 2 1 2 1 2 1 2 1 2 1	4 -3 -5 6 0 -1	Atbilst piemēram uzdevuma tekstā (1.(e) att.). Ir arī citi derīgi atrisinājumi.

Ievaddati	*Izvaddati*	*Piezīme*
6 4 4 2 3 2 4 2 2 2	3 0 6 -1 4	Atbilst 1.(g) attēlam uzdevuma tekstā. Ir arī citi derīgi atrisinājumi.

1. apakšuzdevuma testu ievaddati

Ievaddati
6 5 10 9 6 5 9 8 10 8 3 1

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotais tests	2
2.	$N \leq 2$	8
3.	$N \leq 20$	15
4.	$A \leq 200$	20
5.	$A \leq 1500$	10
6.	$l_i \leq 2$	10
7.	Bez papildu ierobežojumiem	35
	Kopā:	100

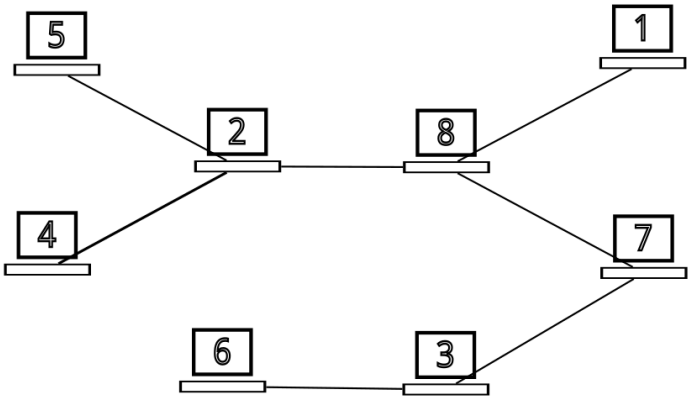
LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
JAUNĀKĀ (8. - 10. KLAŠU) GRUPA

Latvijas informātikas olimpiāde

Bojātais kabelis

Kādā datorfirmā N datori ir saslēgti vienotā tīklā tā, ka katru divu datoru pāris ir vai nu tieši savienots ar kabeli, vai arī eksistē unikāla secīgu datoru virkne no viena datora līdz otram, kur katrus divus blakus datorus savieno kabelis. Datori tīklā ir sanumurēti ar naturāliem skaitļiem no 1 līdz N pēc kārtas.

Viena šāda datortīkla piemērs parādīts 1. attēlā.



1. attēls: Datortīkla piemērs

Meistars Matīss ir pamanījis, ka datortīkls vairs nedarbojas kā nākas, un pēc viņa ilggadējās iepriekšējās pieredzes ir skaidrs, ka vainīgs ir kāds no sistēmā esošajiem kabeļiem.

Meistara rīcībā ir līdzekļi, kas ļauj pieslēgties jebkuriem diviem datoriem un noteikt, vai starp tos savienojošiem kabeļiem kāds nav bojāts.

Uzrakstiet datorprogrammu, kas organizē šādu pieslēgšanos virkni ar mērķi atrast bojāto kabeli.

Komunikācija

Šis ir interaktīvs uzdevums. Jūsu programmai, sākot darbu, pirmajā ievaddatu rindā dots naturāls skaitlis N ($2 \leq N \leq 20000$) – datoru skaits datortīklā. Nākamajā $N-1$ ievaddatu rindā katrā dots viena tieši ar kabeli savienota datoru pāra apraksts – savienoto datoru numuri, kas atdalīti ar tukšumzīmi.

Bojātā kabeļa atrašanās vietu vērtēšanas sistēma tur slepenībā.

Tad jūsu programma var veikt vaicājumus, tos rakstot izvadā sekojošā formātā: «$0\ d_1\ d_2$», kur d_1 un d_2 ir divu datoru numuri – atšķirīgi naturāli skaitļi $(1 \leq d_1, d_2 \leq N)$. Vērtēšanas sistēma uz vaicājumu izdod atbildi nākamajā ievaddatu rindā. Atbilde ir vesels skaitlis – 0, ja kabeļu virknē no d_1 līdz d_2 kāds no kabeļiem ir bojāts, vai 1, ja neviens no kabeļiem šajā virknē nav bojāts.

Jūsu programma katrā testā var veikt ne vairāk kā 20000 vaicājumus.

Kad bojātā kabeļa atrašanās vieta noteikta, programmai jāizvada «$1\ d_x\ d_y$» $\left(1 \leq d_x, d_y \leq N\right)$, kur d_x un d_y ir bojātā kabeļa galos esošo datoru numuri, un darbība jābeidz. Vērtēšanas sistēma neatbildēs uz šo izvadu un nepieņems sekojošus vaicājumus.

Raksturojot kabeli, galapunktu secībai nav nozīmes. Visos testos būs tieši viens bojātais kabelis.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Bojatais**

Piezīmes

Lai nodrošinātu, ka jūsu vaicājumi tiek nodoti vērtēšanas sistēmai, jums ir jāsinhronizē (*flush*) izvada datu plūsma pēc katra vaicājuma:

Valoda	*Piemērs*	*Komentārs*
C++	`std::cout << 0 << " " << p << std::endl;`	"`std::endl`" nodrošina datu plūsmas sinhronizāciju
Go	`fmt.Println(0, p)`	Standarta datu plūsma nav īpaši jāsinhronizē
Java	`System.out.println("0 " + p);` `System.out.flush();`	
Pascal	`writeln('0 ', p);` `flush(output);`	
Python	`print(0, p, flush=True)`	

Ja tiks pārsniegts maksimāli atļautais vaicājumu skaits, var tikt izdots kļūdas paziņojums "Izvaddati nav pareizi".

Izmantojot lietotāja testus sistēmas sadaļā "Testēšana", ievaddatu faila pirmajā rindā jābūt diviem naturāliem skaitļiem – datoru skaitam n un kabeļa kārtas numuram b, kurš tieši ir bojāts. Tad nākamajām $n-1$ rindām jāsatur kabeļu pāru apraksti tādā formātā kā aprakstīts ievaddatos. Lietotāja testu, kas atbilst tekstā dotajam piemēram atrodas sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Piesaistnes“.

Piemērs

Ievaddati	Izvaddati (Jūsu programmas vaicājumi)	Komentāri
8 2 5 8 7 4 2 1 8 3 6 3 7 8 2		Atbilst tekstā dotajam attēlam.
	`0 1 6`	
1		
	`0 7 5`	
1		
	`1 4 2`	Bojātais kabelis var būt vienīgi starp 2 un 4.

Apakšuzdevumi un to vērtēšana

Nr.	**Testu apraksts**	**Punkti**
1.	$N = 2$	1
2.	$N \leq 100$	9
3.	Katrs dators ir savienots ar ne vairāk kā diviem citiem datoriem	20
4.	$N \leq 20000$	70
	Kopā:	100

Ja ir pareizi atrasts bojātais kabelis, tad atkarībā no veikto vaicājumu skaita Q katram testam tiek aprēķināta tā **kvalitāte**. To aprēķina šādi:

- Ja $Q \leq 10001$, tad $\text{kvalitāte} = 1$.
- Ja $10001 < Q \leq 10040$, tad $\text{kvalitāte} = 1 - \frac{Q-10001}{100}$.
- Ja $10040 < Q \leq 20000$, tad $\text{kvalitāte} = 0.6 - \frac{Q-10040}{18000}$
- Ja $20000 < Q$, tad $\text{kvalitāte} = 0$.

Ja grupā ir iekļauti vairāki testi, tad grupas vērtējumu nosaka vissliktāk izpildītais tests (kura kvalitāte ir vismazākā). Par testu grupu piešķirto punktu skaitu aprēķina kā:

$$\text{Piešķirtais punktu skaits} = \text{Sliktākā testa kvalitāte grupā} \cdot \text{Punktu skaits par grupu}$$

LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
VECĀKĀ (11. - 12. KLAŠU) GRUPA

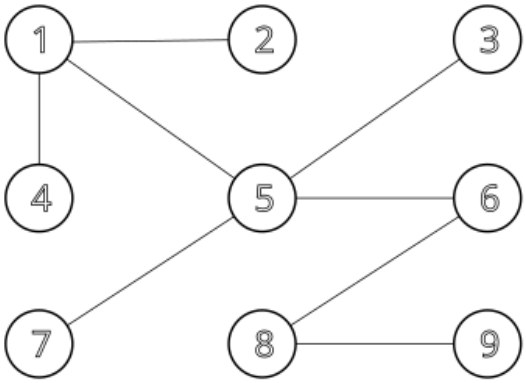
Latvijas informātikas olimpiāde

Dārgākais posms

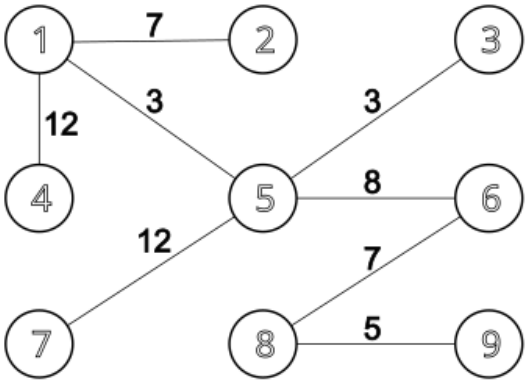
Loģistika ir zinātne par to, kā jebkuras sistēmas ietvaros objektīvāk plānot, regulēt un kontrolēt informācijas, materiālu, produkcijas, cilvēku un enerģijas plūsmas, kā arī visaptveroša šo plūsmu vadīšana; ar ražojumu izplatīšanu saistīto darbu optimizēšana.

Pētnieks Konrāds šobrīd analizē kādas transporta firmas materiālu pārvadāšanas plūsmas. Viņš ir noskaidrojis, ka firma savus pārvadājumus veic tikai starp noteiktām pilsētām tā, ka katrs posms saista tieši divas pilsētas, turklāt pārvadājums no vienas pilsētas līdz kādai citai vienmēr ir šādu posmu virkne, kas vienmēr iespējama tikai vienā vienīgā veidā. Pilsētas ir sanumurētas ar naturāliem skaitļiem no 1 līdz N pēc kārtas.

Vienas šādas sistēmas ar pilsētas savienojošajiem posmiem piemērs parādīts 1. attēlā.



1. attēls: Pilsētas un tās savienojošie posmi (zināmā struktūra)



2. attēls: Pilsētas un tās savienojošie posmi (nezināmās posmu izmaksas)

Konrāds ir noskaidrojis, ka pārvadājumu izmaksas katrā posmā var izteikt kā naturālu skaitli, kas nepārsniedz kādu zināmu vērtību S_m, un tās dažādiem posmiem var atšķirties. Viņš vēlas noteikt, kurā posmā izmaksas ir vislielākās. Diemžēl firma nevēlas sniegt pilnu informāciju par esošajām izmaksām, bet ir gatava sniegt atbildi „jā" vai „nē" uz šādiem vaicājumiem: „Vai maršrutā no pilsētas X līdz pilsētai Y kādā posmā pārvadājumu izmaksas ir vismaz S?"

Uzrakstiet datorprogrammu, kas ģenerē šādu vaicājumu virkni ar mērķi atrast posmu ar vislielākajām pārvadājumu izmaksām!

Komunikācija

Šis ir interaktīvs uzdevums. Jūsu programmai, sākot darbu, pirmajā ievaddatu rindā doti divi naturāli skaitļi N (pilsētu skaits, $2 \leq N \leq 5000$) un S_m (maksimālās iespējamās viena posma izmaksas, $S_m \leq 10^9$), kas atdalīti ar tukšumzīmi. Nākamajā $N-1$ ievaddatu rindā katrā dots viena divas pilsētas savienojošā posma apraksts – savienoto pilsētu numuri, kas atdalīti ar tukšumzīmi.

Dārgākā posma (vai posmu) atrašanās vietu un precīzās tā izmaksas vērtēšanas sistēma tur slepenībā.

Tad jūsu programma var veikt vaicājumus, tos rakstot izvadā sekojošā formātā: «$0\ p_1\ p_2\ s$», kur p_1 un p_2 ir divu pilsētu numuri – atšķirīgi naturāli skaitļi $(1 \leq p_1, p_2 \leq N)$, bet s $(1 \leq s \leq S_m)$ – izmaksas, izteiktas kā naturāls skaitlis. Vērtēšanas sistēma uz vaicājumu izdod atbildi nākamajā ievaddatu rindā. Atbilde ir vesels skaitlis – 0, ja maršrutā no p_1 līdz p_2 visu posmu izmaksas ir mazākas par s, vai 1, ja vismaz viena posma izmaksas ir lielākas vai vienādas ar s.

Jūsu programma katrā testā var veikt ne vairāk kā 150000 vaicājumus.

Kad dārgākā posma atrašanās vieta noteikta, programmai jāizvada «$1\ p_x\ p_y$» $(1 \leq p_x, p_y \leq N)$, kur p_x un p_y ir dārgākā posma galos esošo pilsētu numuri, un darbība jābeidz. Vērtēšanas sistēma neatbildēs uz šo izvadu un nepieņems sekojošus vaicājumus.

Ja sistēmā ir vairāki posmi ar dārgākajām izmaksām, jāizvada informācija par jebkuru no tiem. Raksturojot posmu, pilsētu numuru secībai nav nozīmes.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Dargakais**

Piezīmes

Lai nodrošinātu, ka jūsu vaicājumi tiek nodoti vērtēšanas sistēmai, jums ir jāsinhronizē (*flush*) izvada datu plūsma pēc katra vaicājuma:

Valoda	*Piemērs*	*Komentārs*
C++	`std::cout << 0 << " " << p << std::endl;`	“`std::endl`” nodrošina datu plūsmas sinhronizāciju
Go	`fmt.Println(0, p)`	Standarta datu plūsma nav īpaši jāsinhronizē
Java	`System.out.println("0 " + p);` `System.out.flush();`	
Pascal	`writeln('0 ', p);` `flush(output);`	
Python	`print(0, p, flush=True)`	

Ja tiks pārsniegts maksimāli atļautais vaicājumu skaits, var tikt izdots kļūdas paziņojums “Izvaddati nav pareizi”.

Izmantojot lietotāja testus sistēmas sadaļā “Testēšana”, ievaddatu faila pirmajā rindā jābūt diviem naturāliem skaitļiem – pilsētu skaitam n un maksimālajām viena posma izmaksām S_m. Tad nākamajām $n-1$ rindām jāsatur savienojošo posmu apraksti formātā «u v w», kur u un v ir posma galos esošās pilsētas, bet w - šī posma izmaksas. Lietotāja tests, kas atbilst tekstā dotajam piemēram, atrodas sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Piesaistnes“.

Piemērs (atbilst 2. att.)

Ievaddati	Izvaddati (Jūsu programmas vaicājumi)	Komentāri
9 16 2 1 8 9 4 1 1 5 5 6 5 7 8 6 3 5		Atbilst 1. attēlam.
	0 2 9 10	
0		
	0 4 7 10	
1		
	0 7 4 13	
0		
	0 7 4 12	
1		
	0 5 3 12	
0		
	0 4 1 12	
1		
	1 1 4	Visdārgākais posms ir starp 1 un 4. Tikpat dārgs, bet ne dārgāks, pēc šī dialoga rezultātiem var būt posms starp 5 un 7.

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	$N = 2$	1
2.	Katra pilsēta ir savienota ar ne vairāk kā 2 citām pilsētām	10
3.	$S_m = 2$	10
4.	$S_m = 256$	15
5.	Bez papildus ierobežojumiem	64
	Kopā:	100

Ja ir pareizi atrasts bojātais kabelis, tad atkarībā no veikto vaicājumu skaita Q katram testam tiek aprēķināta tā **kvalitāte**. To aprēķina šādi:

- Ja $Q \leq 3500$, tad $\text{kvalitāte} = 1$.
- Ja $3500 < Q \leq 10000$, tad $\text{kvalitāte} = 1 - \frac{Q-3500}{10000}$.
- Ja $10000 < Q \leq 50000$, tad $\text{kvalitāte} = 0.35 - \frac{Q-10000}{400000}$
- Ja $50000 < Q \leq 150000$, tad $\text{kvalitāte} = 0.25 - \frac{Q-50000}{500000}$
- Ja $150000 < Q$, tad $\text{kvalitāte} = 0$.

Ja grupā ir iekļauti vairāki testi, tad grupas vērtējumu nosaka vissliktāk izpildītais tests (kura kvalitāte ir vismazākā). Par testu grupu piešķirto punktu skaitu aprēķina kā:

$$\text{Piešķirtais punktu skaits} = \text{Sliktākā testa kvalitāte grupā} \cdot \text{Punktu skaits par grupu}$$

LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
JAUNĀKĀ (8. - 10. KLAŠU) GRUPA

Latvijas informātikas olimpiāde

Izsmalcinātās kūkas

Izsmalcinātais ēdiena kritiķis Valters ir ieradies īpaši izsmalcinātā beķerejā. Beķeris viņam priekšā noliek n kūkas secīgi rindā, kur i-tās kūkas garšīgums ir g_i un tās tips ir t_i.

Valters pussekundi pēc kūku ieraudzīšanas pavēstīja, ka savā pasūtījumā vēlas daudzveidību, tāpēc ir gatavs nogaršot tikai tādus kūku komplektus, kur katrām divām kūkām to tipu vērtības atšķiras par vismaz k (formāli, katrām divām paņemtajām kūkām i, j $(i \neq j)$ jāizpildās $|t_i - t_j| \geq k$).

Beķeris grib atstāt labu iespaidu uz ēdiena kritiķi Valteru, tāpēc no visām kūkām izvēlēsies tādu komplektu, kas atbilst Valtera nosacījumam, kā arī to garšīgumu summa ir vislielākā.

Uzrakstiet datorprogrammu, kas atrod vislielāko garšīguma summu!

Ievaddati

Pirmajā rindā ir doti divi naturāli skaitļi n un k $(1 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 10^9)$.

Nākamajās n rindās katrā doti divi veseli skaitļi g_i un t_i $(1 \leq g_i, t_i \leq 10^9)$ – i-tās kūkas garšīgums un tips.

Starp katriem diviem blakus skaitļiem ievaddatos ir tukšumzīme.

Izvaddati

Izvaddatu vienīgajā rindā jābūt vienam veselam skaitlim – maksimālajai garšīgumu summai.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Kuka**

Piemēri

Ievaddati	*Izvaddati*	*Piezīme*
5 3 6 10 2 3 5 5 4 8 2 7	11	Beķeris izvēlējās pēc kārtas pirmo un trešo kūku un ieguva garšīguma summu 6+5=11. Šāda izvēle atbilst Valtera nosacījumam, jo šo kūku tipi |10-5|≥3.

Ievaddati	*Izvaddati*
7 5 13 8 1 12 9 14 2 12 8 3 15 4 5 3	30

1. apakšuzdevuma testu ievaddati

Ievaddati
8 7 6 9 15 9 15 8 4 3 9 15 10 15 14 14 5 15

Ievaddati
6 5 6 5 9 10 10 10 14 16 19 10 7 6

Ievaddati
7 2 12 15 5 12 14 8 19 1 8 5 4 11 3 2

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotie trīs testi	2
2.	$n \leq 18$	10
3.	Ja $g_i \leq g_j$ $(i \neq j)$, tad arī izpildās $2g_i \leq g_j$	18
4.	$n \leq 2000$	21
5.	$\lvert t_i - t_j \rvert \geq \frac{k}{2}$ katram i, j $(i \neq j)$ pārim	23
6.	Bez papildu ierobežojumiem	26
	Kopā:	100

LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
VECĀKĀ (11. - 12. KLAŠU) GRUPA

Latvijas informātikas olimpiāde

Paralelogrami

Šajā uzdevumā aplūkosim paralelogramus, kuru virsotnes atrodas Dekarta koordinātu plaknē punktos ar veselām koordinātām. Atkarībā no paralelograma virsotņu novietojuma, arī tā diagonāļu krustpunkts var atrasties punktā ar veselām koordinātām.

Piemēram, ja paralelograma virsotnes atrodas punktos ar koordinātām $(-4;-2)$, $(-2;2)$, $(8;4)$ un $(6;0)$, tad tā diagonāļu krustpunkts atrodas punktā ar koordinātām $(2;1)$.

Uzrakstiet datorprogrammu, kas nosaka, cik dažādos veidos no dotajiem N punktiem ar veselām koordinātām var izvēlēties četrus punktus tā, ka šī paralelograma diagonāles krustojas koordinātu sākumpunktā $(0;0)$.

Ievaddati

Pirmajā rindā dots punktu skaits – naturāls skaitlis $N (4 \leq N \leq 5 \cdot 10^5)$. Nākamajās N rindās katrā dotas viena punkta koordinātas (abscisa un ordināta) – divi veseli skaitļi x_i un y_i, kur $(-10^9 \leq x_i, y_i \leq 10^9)$. Starp skaitļiem ievaddatos ir tukšumzīme. Visi dotie plaknes punkti ir savā starpā atšķirīgi.

Izvaddati

Izvaddatu vienīgajā rindā jābūt veselam nenegatīvam skaitlim – atšķirīgo veidu skaitam, kā no N dotajiem punktiem izvēlēties četrus ar aprakstītajām īpašībām. Punktu četrinieki, kuros atšķiras tikai punktu secība, netiek uzskatīti par atšķirīgiem.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Paralelogrami**

Piemēri

Ievaddati	*Izvaddati*	*Piezīme*
8 -4 -2 -2 2 8 4 6 0 4 2 2 -2 -6 0 -8 -4	5	Šie paralelogrami ir: 1: (-8;-4),(-2;2),(8;4),(2;-2) 2: (-8;-4),(-6;0),(8;4),(6;0) 3: (-6;0),(-2;2),(6;0),(2;-2) 4: (-4;-2),(-2;2),(4;2),(2;-2) 5: (-4;-2),(-6;0),(4;2),(6;0). Ievērojiet, ka neder punktu četrinieks (-8;-4), (-4;-2),(8;4),(4;2), jo šāda četrstūra pretējās malas nav paralēlas - tām ir kopīgi punkti.

Ievaddati	*Izvaddati*
4 -1 -1 2 2 -1 2 2 -1	0

1. apakšuzdevuma testa ievaddati

Ievaddati
11 2 6 1 3 3 3 -1 2 2 2 -1 -1 -2 -2 1 -2 -1 -3 2 -4 -2 -6

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotais tests	2
2.	$4 \leq N \leq 100$	20
3.	$-10^4 \leq x_i, y_i \leq 10^4$	20
4.	$101 \leq N \leq 1000$	28
5.	Bez papildu ierobežojumiem	30
	Kopā:	100

LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE
VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS
JAUNĀKĀ (8. - 10. KLAŠU) GRUPA

Latvijas informātikas olimpiāde

Siers

Pelēns Lio ir atradis milzīgu siera šķēli, ko var raksturot kā rūtiņu laukumu ar n rindām un m kolonnām. Siera šķēles rūtiņu i-tajā rindā un j-tajā kolonnā var aprakstīt ar veselu skaitli a_{ij} - tā garšīgumu. Kādas siera rūtiņas var būt sapelējušas – šādu rūtiņu garšīgums tiek uzdots kā negatīvs skaitlis.

Lio vēlas šo sieru ēst pēc sekojošiem noteikumiem:

- katrā rindā jāapēd vismaz viena rūtiņa;
- nevienā rindā nedrīkst apēst visas rūtiņas;
- tieši k rindās viņam siers jāēd sākot no kreisās puses (pēc kārtas rūtiņas ar indeksiem $1, 2, ..., i$), pārējās $n - k$ rindās viņam siers jāēd sākot no labās puses (pēc kārtas rūtiņas ar indeksiem $m, m - 1, ..., i$).

Lio vēlas, lai visu apēsto rūtiņu garšīgumu summa būtu vislielākā.

Uzrakstiet datorprogrammu, kas aprēķina un izvada šo vērtību!

Ievaddati

Pirmajā rindā doti siera laukuma izmēri – naturāli skaitļi n un m ($1 \leq n \leq 10^5, 2 \leq m \leq 10^5$, $n \cdot m \leq 5 \cdot 10^5$).

Otrajā rindā dots naturāls skaitlis k ($0 \leq k \leq n$).

Nākamajās n rindās katrā doti m veseli skaitļi a_{ij} ($|a_{ij}| \leq 10^9$). Katram $i (1 \leq i \leq n)$ un $j (1 \leq j \leq m)$ i-tās rindas j-tās kolonnas rūtiņas garšīgums dots ievaddatu $i + 2$-ajā rindā kā j-tais skaitlis pēc kārtas.

Starp katriem diviem blakus skaitļiem ievaddatos ir tukšumzīme.

Izvaddati

Izvaddatu vienīgajā rindā jābūt vienam veselam skaitlim – maksimālajai garšīgumu summai.

Ierobežojumi un prasības

Atmiņas apjoma un izpildes laika ierobežojumus skatīt sacensību sistēmā uzdevuma sadaļā „Formulējums“ ⇒ „Tehniskā informācija“.

Klases vārds valodā Java rakstītam risinājumam: **Siers**

Piemērs

Ievaddati	*Izvaddati*	*Piezīme*
3 3 2 2 0 1 -1 3 -1 2 -2 1	6	Lio siers jāēd no kreisās puses tieši divās rindās. Maksimālo garšīgumu summu var iegūt, apēdot vienu vai divas rūtiņas no 1. rindas kreisās puses, divas rūtiņas no 2. rindas labās puses un vienu rūtiņu no 3. rindas kreisās puses. Iegūtā garšīgumu summa ir 2+(-1)+3+2=6. Tā kā Lio nevienā rindā nedrīkst apēst visas rūtiņas, tad lielāku garšīgumu summu iegūt nav iespējams.

1. apakšuzdevuma testa ievaddati

Ievaddati
4 6 3 2 2 1 0 -1 2 2 5 2 -4 -5 2 2 -2 -4 -1 -1 2 1 -4 5 5 2 -5

Apakšuzdevumi un to vērtēšana

Nr.	Testu apraksts	Punkti
1.	Uzdevuma tekstā dotais tests	2
2.	$n = 1$	11
3.	$k = n$	18
4.	$m = 2, n \leq 10$	20
5.	$1000 \leq n \leq 2000$	21
6.	Bez papildu ierobežojumiem	28
	Kopā:	100