

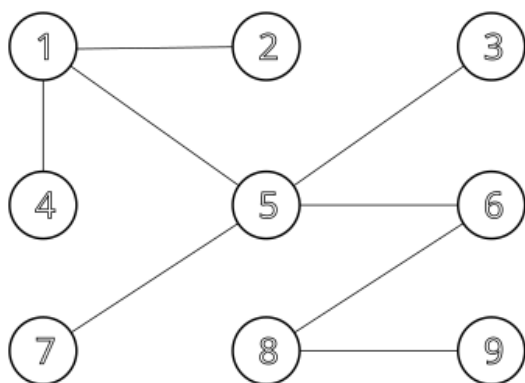
LATVIJAS 38. INFORMĀTIKAS OLIMPIĀDE  
 VALSTS OLIMPIĀDES PIRMĀ DIENA - 2025. GADA 27. FEBRUĀRIS  
 VECĀKĀ (11. - 12. KLAŠU) GRUPA

## Dārgākais posms

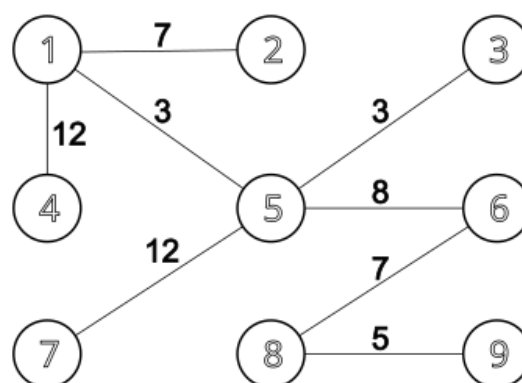
Loģistika ir zinātne par to, kā jebkuras sistēmas ietvaros objektīvāk plānot, regulēt un kontrolēt informācijas, materiālu, produkcijas, cilvēku un enerģijas plūsmas, kā arī visaptveroša šo plūsmu vadīšana; ar ražojumu izplatīšanu saistīto darbu optimizēšana.

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 izvadā 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 jāsinchronizē (*flush*) izvada datu plūsma pēc katra vaicājuma:

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

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 jāsaturo 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
<b>Kopā:</b>		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 kvalitāte = 1.
- Ja  $3500 < Q \leq 10000$ , tad kvalitāte =  $1 - \frac{Q-3500}{10000}$ .
- Ja  $10000 < Q \leq 50000$ , tad kvalitāte =  $0.35 - \frac{Q-10000}{40000}$ .
- Ja  $50000 < Q \leq 150000$ , tad kvalitāte =  $0.25 - \frac{Q-50000}{50000}$ .
- Ja  $150000 < Q$ , tad 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:

Piešķirtais punktu skaits = Sliktākā testa kvalitāte grupā · Punktu skaits par grupu