

Uzdevums: TOW

Tornis



BOI 2025, diena 1. Atmiņas ierobežojums: 256 MB.

2025.04.26

Par Toruņas šķībo torni kļūst daudzas leģendas. Torņa siena ir aplis ar $N \geq 3$ vienmērīgi izvietotām durvīm (citiem vārdiem, durvis atrodas regulāra N -stūra virsotnēs). Durvis ir numurētas no 0 līdz $N - 1$ **nejaušā secībā**. Sīkāk skatīt vērtēšanas sadaļu.

Viena mazāk zināma leģenda stāsta, ka katram jaunajam torņa iemītniekam jāizpilda īpašs pārbaudījums. Pārbaudījuma uzdevums ir nosaukt visas durvis, sākot ar kādām durvīm un virzoties pa apli (pulksteņrādītāja virzienā vai pretējā), katras durvis apmeklējot tieši vienu reizi.

To jāizdara, pašu torni neredzot. Tā vietā jaunais iemītnieks drīkst uzdot vaicājumus šādā formā: “Dotas trīs dažādas durvis x, y, z . Kurš durvju pāris — $\{x, y\}$, $\{y, z\}$ vai $\{z, x\}$ — atrodas vistuvāk viens otram?” Atbildē tiek norādīti visi pāri (starp $\{x, y\}$, $\{y, z\}$ un $\{z, x\}$), kuriem Eiklīda attālums (*Euclidean distance*) ir vismazākais. Attālums ir īsākā nogriežņa garums starp konkrētajām durvīm.

Jūsu uzdevums ir uzrakstīt programmu, kas vaicās mazu skaitu ar šādiem jautājumiem, lai noskaidrotu durvju secību.

Komunikācija

Šis ir interaktīvs uzdevums. Jums jāuzraksta programma, kas atrod korektu šī uzdevuma risinājumu un komunicē ar žūrijas programmu (**interactor**), lasot no standarta ievades un rakstot uz standarta izvadi.

Komunikācijas sākumā programma nolasa divus veselus skaitļus t un k ($1 \leq t \leq 100$, $1 \leq k \leq 12\,000$) — testa gadījumu skaitu un maksimāli pieļaujamo vidējo vaicājumu skaitu. Sīkāk par k skatīt vērtēšanas sadaļu.

Katra testa gadījuma sākumā programmai jānolasa vienu veselu skaitli n ($3 \leq n \leq 500$) — durvju skaitu tornī.

Tālāk programmai vaicājumi jājautā šādi:

- Programmai jāizvada vienu rindu formā

? x y z

uz standarta izvadi, kur x, y, z ir trīs dažādi veseli skaitļi ($0 \leq x, y, z \leq n - 1$). Šī rinda attēlo vienu jautājumu par durvīm x, y, z .

- Atbilde tiks dota šādi:

r
 a_1 b_1
...
 a_r b_r

kur r ir vesels skaitlis ($1 \leq r \leq 3$) — pāru skaits, kuri atrodas vistuvāk; katru pāri raksturo divi vesēlie skaitļi a_i un b_i ($a_i, b_i \in \{x, y, z\}$ un $a_i < b_i$).

Kad durvju secība ir noskaidrota, izvadiet vienu rindu formā

! x_0 x_1 ... x_{n-1}

kur x_0, x_1, \dots, x_{n-1} ir durvju secība. Eksistē tieši $2n$ korektu atbilžu, jo var sākt ar jebkurām durvīm un virzīties vienā vai otrā virzienā. Jebkura no tām tiks pieņemta.

Paturiet prātā, ka pēc katra vaicājuma vai atbildes jums ir jāsinhronizē (*flush*) izvades buferi izmantojot `cout.flush()` (vai `fflush(stdout)`), ja izmantojat `printf`) C++ valodā vai jāpievieno `flush = True` argumentu `print` komandai Python valodā. Citādi jūsu programma var saņemt `Time Limit Exceeded` vērtējumu.

Pēc atbildes izvadīšanas programmai nekavējoties jāpāriet pie nākamā testa gadījuma vai jābeidz darbība, ja testa gadījumi ir beigušies.

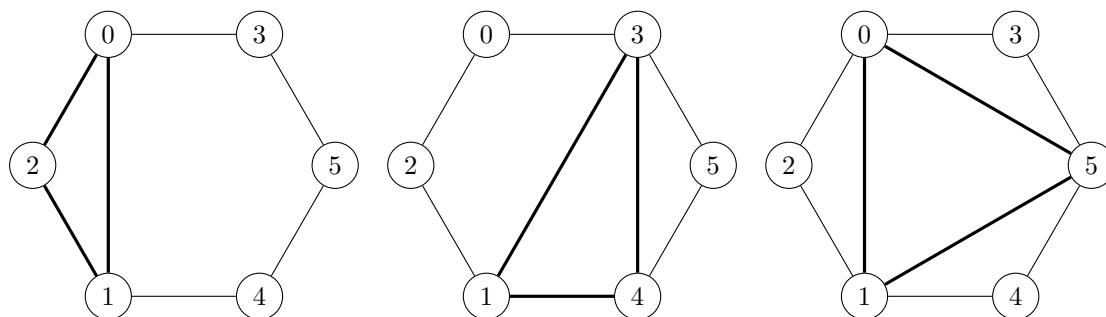
Jūsu programma nedrīkst vērt valā jebkādas failus vai izmantot citus resursus. Tā var izmantot standarta kļūdas izvadi atklādošanas nolūkiem, bet jāņem vērā, ka rakstīšana uz šo plūsmu patērē laiku.

Žūrijas programma (*interactor*) nav adaptīva — katra testa sākumā durvju secība ir nofiksēta un komunikācijas gaitā nemainās.

Komunikācijas piemērs

Pieņemsim, ka mums ir viens testa gadījums ar $n = 6$ un durvju secību 5, 3, 0, 2, 1, 4. Komunikācija varētu izskatīties šādi:

Žūrijas programma	Jūsu programma	Komentārs
1 100		$t = 1, k = 100$.
6		Žūrijas programma ievada durvju skaitu.
	? 0 1 2	Jūsu programma jautā, kurš pāris ir tuvāk.
2 0 2 1 2		Tuvākie durvju pāri ir $\{0, 2\}$ un $\{1, 2\}$.
	? 4 1 3	Jūsu programma jautā, kurš pāris ir tuvāk.
1 1 4		Tuvākie durvju pāri ir $\{1, 4\}$.
	? 0 5 1	Jūsu programma jautā, kurš pāris ir tuvāk.
3 0 5 0 1 1 5		Tuvākie durvju pāri ir $\{0, 5\}$, $\{0, 1\}$ un $\{1, 5\}$.
	! 4 5 3 0 2 1	Jūsu programma izvada korektu durvju secību.



Piemēru paskaidrojums:

Attēlos redzamas durvis ar saviem atbilstošajiem numuriem gar torņa sienu. Pirmajā attēlā (pa kreisi) attēlots trīsstūris ar durvīm 0, 1, 2. Tas atbilst pirmajam jūsu programmas vaicājumam. Var redzēt, ka pāri $\{0, 2\}$ un $\{1, 2\}$ ir vistuvāk. Otrajā attēlā (pa vidu) attēlots trīsstūris 1, 4, 3, atbilstot otrajam vaicājumam. Tuvākais pāris ir $\{1, 4\}$. Trešajā attēlā (pa labi) attēlots trīsstūris 0, 1, 5, atbilstot trešajam vaicājumam. Labajā zīmējumā trīsstūris 0, 1, 5 — trešajam vaicājumam. Acīmredzami, ka visi durvju pāri ir vienādi tuvi.

Šajā gadījumā korektas atbildes būtu virknes 0, 2, 1, 4, 5, 3 un 5, 4, 1, 2, 0, 3 (un vēl dažas citas).

Vērtēšana

Vērtēšana ir sadalīta apakšuzdevumos. Katram apakšuzdevumam ir tieši viens tests, kas satur $t = 100$ testa gadījumus. Katram testam vidējais vaicājumu skaits, ko vaicāja jūsu programma, tiek aprēķināts ņemot kopējo vaicājumu skaitu pa visiem testa gadījumiem un izdalot to ar testa gadījumu skaitu. Ja šis vidējais pārsniedz k konkrētam apakšuzdevumam, jūs par attiecīgo apakšuzdevumu saņemsiet 0 punktus. Citādi, par apakšuzdevumiem no 1 līdz 4 jūs saņemsiet visus punktus par attiecīgo apakšuzdevumu.

Par pēdējo apakšuzdevumu jūsu punkti tiks aprēķināti sekojoši. Ar k^* apzīmēsim veikto vaicājumu vidējo vērtību, ko jūsu programma jautāja. Tad punkti tiek aprēķināti pēc šādas formulas:

$$\left\lceil 56 \cdot \min\left(1, \frac{12000 - k^*}{7800}\right) \right\rceil.$$

Tas nozīmē, ka punkti lineāri pieaug no 0 līdz 56, k^* samazinoties no 12000 līdz 4200.

Lūdzu, ņemiet vērā, ka, ja jūsu programma kaut vienā testa gadījumā izdod nepareizu atbildi, jūs saņemsiet 0 punktus par attiecīgo apakšuzdevumu neatkarīgi no vaicājumu skaita.

Papildus ierobežojumi apakšuzdevumiem:

Apakšuzdevums	Ierobežojumi	Punkti
1	$k = 8000, 3 \leq n \leq 9$	6
2	$k = 4500, 40 \leq n \leq 50$	7
3	$k = 3000, 90 \leq n \leq 100$	9
4	$k = 4500, n = 400$, eksistē korekta atbilde x_0, \dots, x_{n-1} , kur $x_i = i$ visiem $200 \leq i \leq 399$	22
5	$k = 12000, n = 500$	līdz 56

Turklāt jūs varat pieņemt, ka katrs testa gadījums ir ģenerēts vispirms izvēloties n ar **vienādu varbūtību** no visām n vērtībām, kas atbilst apakšuzdevuma ierobežojumiem, un pēc tam nejauši izvēloties durvju secību ar **vienādu varbūtību** no visām n durvju secībām, kas atbilst apakšuzdevuma ierobežojumiem.