

# Kur ir sakne?

Šis ir interaktīvais uzdevums

Ir dots  $n$  virsotņu koks. Koks ir grafs, kurā katru virsotņu pāri savieno tieši viens ceļš. **Ir garantēts, ka vismaz viena virsotne ir savienota ar vismaz 3 citām virsotnēm.** Viena virsotne ir sakne. Uzdevums ir noteikt, kura virsotne tā ir.

Ir atļauts uzdot vaicājumus šādā formātā:

- Dotajai virsotņu kopai  $a_1, a_2, \dots, a_m$  pārbaudīt, vai dotajā kopā atrodas šo virsotņu zemākais kopīgais sencis (vecāks).

Paskaidrojums. Virsotņu kopas zemākais kopīgais sencis (vecāks) ir no saknes vistālāk esošā virsotne  $v$ , kura pieder visiem vienkāršajiem ceļiem no saknes uz katru virsotni.

## Interaktivitāte

Interaktivitāte sākas ar viena vesela skaitļa  $n$  ( $4 \leq n \leq 500$ ) nolasīšanu.  $n$  - virsotņu skaits.

Pēc tam jānolasa nākamās  $n - 1$  rindas.  $i$ -tajā rindā ir doti divi veseli skaitļi  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ), kas norāda, ka starp virsotnēm  $a_i, b_i$  ir šķautne.

Ir garantēts, ka šīs  $n - 1$  šķautnes veido koku un ka eksistē vismaz viena virsotne, kas ir savienota ar vismaz 3 citām virsotnēm.

Lai uzdotu jautājumu, vispirms jāizvada "?". Pēc tam vesels skaitlis  $m$ , un pēc tam  $m$  atšķirīgi veseli skaitļi  $a_1, a_2, \dots, a_m$  ( $1 \leq m \leq n$ ,  $1 \leq a_i \leq n$ , visi  $a_i$  ir atšķirīgi). Šie skaitļi ir virsotnes, kurām jāpārbauda, vai to sastāvā ir arī to zemākais kopīgais sencis (vecāks).

Tiks sniegta atbilde "YES", ja to zemākais kopīgais sencis (vecāks) ir viena no virsotnēm  $a_1, a_2, \dots, a_m$ . Pretējā gadījumā tiks sniegta atbilde "NO".

Var uzdot ne vairāk kā 1000 vaicājumus. Atbildes izvadīšana netiek skaitīta kā vaicājums.

Kad ir noteikta saknes virsotne, ir jāizvada simbols "!", un pēc tā viens vesels skaitlis  $v$  ( $1 \leq v \leq n$ ) - saknes virsotne. Pēc tam programmai ir jābeidz darbs.

Pēc vaicājuma izvadīšanas nedrīkst aizmirst izvadīt pāreju jaunā rindā un iztīrīt izvades buferi. Lai to izdarītu, ir jāizmanto

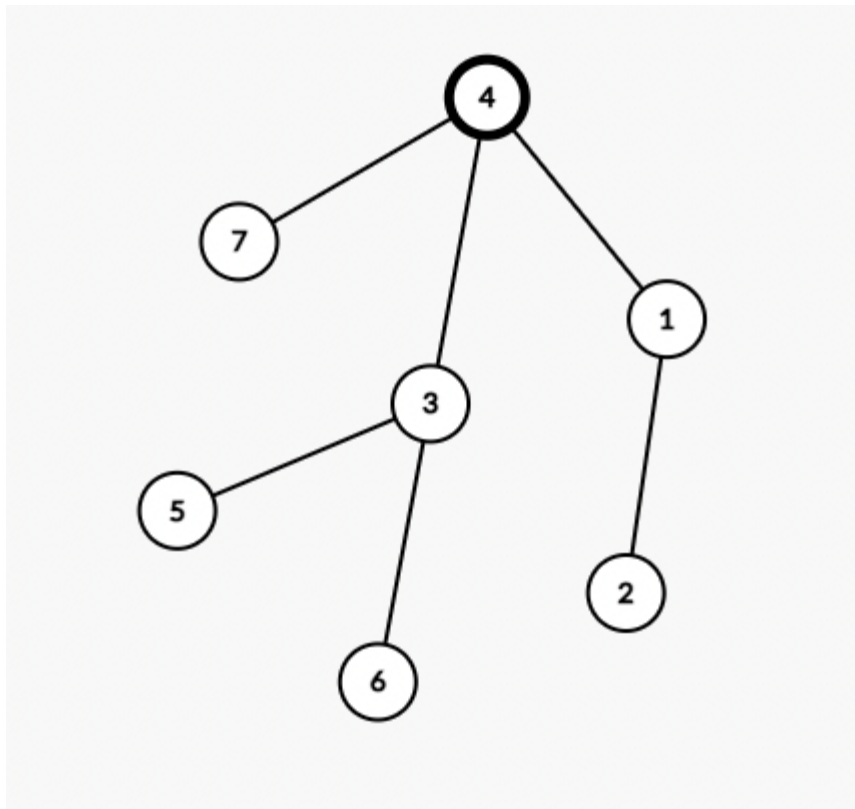
- `fflush(stdout)` vai `cout.flush()` valodā C++;
- `stdout.flush()` valodā Python;

Ir garantēts, ka katrā testā pirms interaktivitātes sākuma koks un tā sakne ir fiksēti un paliek nemainīgi testa izpildes laikā. Citiem vārdiem sakot, **interaktivitāte nav adaptīva**.

## Piemērs

```
Ievaddati:
7
4 1
1 2
4 3
3 5
3 6
4 7
Izvaddati:
? 2 5 6
Ievaddati:
NO
Ievaddati:
? 3 6 3 5
Ievaddati:
YES
Ievaddati:
? 2 1 7
Ievaddati:
NO
Ievaddati:
? 2 4 6
Ievaddati:
YES
Ievaddati:
! 4
```

# Piezīme



Slepenā sakne ir virsotne Nr. 4.

Pirmajā vaicājumā virsotņu 5 un 6 zemākais kopīgais sencis (vecāks) ir virsotne 3, kas nav iekļauta vaicājuma kopā. Tādēļ atbilde ir "NO".

Otrajā vaicājumā virsotņu 3, 5 un 6 zemākais kopīgais sencis (vecāks) ir virsotne 3. Tādēļ atbilde ir "YES".

Trešajā vaicājumā virsotņu 1 un 7 zemākais kopīgais sencis (vecāks) ir virsotne 4. Tādēļ atbilde ir "NO".

Ceturtajā vaicājumā virsotņu 4 un 6 zemākais kopīgais sencis (vecāks) ir virsotne 4. Tādēļ atbilde ir "YES".

Pēc tam var izdarīt minējumu, ka saknes virsotne ir virsotne Nr. 4. Tā ir pareizā atbilde.

## Vērtēšana

Var uzdot ne vairāk kā 1000 vaicājumu.

1. (7 punkti):  $n \leq 9$

2. (10 punkti):  $n \leq 30$

3. (līdz 83 punktiem):  $n \leq 500$

Trešajā apakšuzdevumu grupā, ja  $k \leq 9$ , var saņemt 83 punktus. Citādi var saņemt  $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$  punktus.

Tādējādi, lai iegūtu maksimālo punktu skaitu, risinājumā katrā apakšuzdevuma testā ir jāzīmanto ne vairāk kā 9 vaicājumi.

C++ kods, kas aprēķina punktu skaitu  $((k \leq 9) ? 83 : \max(10, \text{int}(83 * (1 - \log(k - 6.0) / 7))))$