

# Puu juure leidmine

See on interaktiivne ülesanne

Sulle on antud  $n$  tipuga puu. Puu on selline graaf, milles iga kahe tipu vahel on täpselt üks iseenastav mitteloikav tee. **On teada, et selles puus on vähemalt üks tipp, millel on vähemalt kolm naabrit.** Üks puu tippudest on juurtipp ja sinu ülesanne on see üles leida. Juure leidmiseks võid sa esitada järgmist tüüpi päringuid:

- Kontrollida antud tippude hulga  $a_1, a_2, \dots, a_m$  kohta, kas nende lähim ühine eellane on nende hulgas.

Tipp  $v$  on tippude hulga  $S$  ühine eellane, kui teed kõigist hulka  $S$  kuuluvatest tippudest juurtippu lähevad läbi tipu  $v$ . Tippude hulga  $S$  lähim ühine eellane (*lowest common ancestor*, LCA) on nende ühiste eellaste hulgast selline, mis on juurtipust kõige kaugemal.

## Suhtlus

Sinu programm peab töö alguses lugema sisendist puu tippude arvu  $n$  ( $4 \leq n \leq 500$ ).

Seejärel peab programm lugema veel  $n - 1$  rida. Nende hulgas  $i$ . real on täisarvud  $a_i$  ja  $b_i$  ( $1 \leq a_i, b_i \leq n$ ), mis näitavad, et puus on serv tippude  $a_i$  ja  $b_i$  vahel.

On teada, et need  $n - 1$  serva moodustavad puu ja selles on vähemalt üks tipp, millel on vähemalt kolm naabrit.

Päringu esitamiseks väljasta kõigepealt märk "?", seejärel täisarv  $m$  ja siis  $m$  paarikaupa erinevat täisarvu  $a_1, a_2, \dots, a_m$  ( $1 \leq m \leq n$ ,  $1 \leq a_i \leq n$ ,  $a_i$  peavad olema paarikaupa erinevad): nende tippude numbrid, mille kohta sa tahad kontrollida, kas nende lähim ühine eellane on nende hulgas.

Vastuseks kirjutab testimiskeskkond "YES", kui tippude  $a_1, a_2, \dots, a_m$  lähim ühine eellane on nende tippude endi hulgas, või "NO", kui ei ole.

Igas testis võid esitada kuni 1 000 päringut, aga sinu skoor sõltub sellest, kui palju päringuid sa kasutad. (Täpsem info selle kohta on lõigus Hindamine.) Vastuse väljastamine päringuna arvesse ei lähe.

Kui sa oled juurtipu kindlaks teinud, väljasta kõigepealt märk "!" ja siis üks täisarv  $v$  ( $1 \leq v \leq n$ ): juurtipu number. Selle järel peab programm töö lõpetama.

Pea meeles, et iga päringu järel tuleb väljastada reavahetus ja tühjendada väljundpuhver; väljundpuhvi tühjendamine:

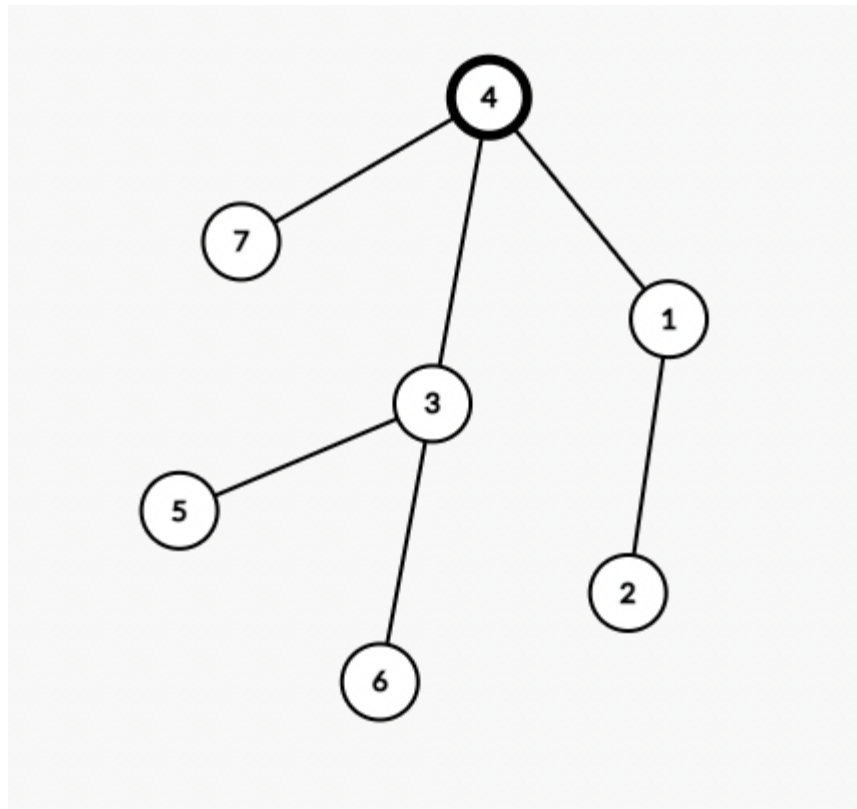
- C++'is: `fflush(stdout)` või `cout.flush()`
- Pythonis: `stdout.flush()`

On teada, et igas testis on puu kuju ja selle juurtipp fikseeritud enne sinu programmiga suhtlemise algust. See tähendab, et **testimine ei ole adaptiivne**.

## Näide

```
Sisend:
7
4 1
1 2
4 3
3 5
3 6
4 7
Väljund:
? 2 5 6
Sisend:
NO
Väljund:
? 3 6 3 5
Sisend:
YES
Väljund:
? 2 1 7
Sisend:
NO
Väljund:
? 2 4 6
Sisend:
YES
Väljund:
! 4
```

# Selgitus



Selles näites on juurtipp number 4.

Esimeses päringus on tippude 5 ja 6 lähim ühine eellane tipp 3, mis ei ole tippude 5 ja 6 hulgas, seega on vastus sellele päringule "NO".

Teises päringus on tippude 3, 5 ja 6 lähim ühine eellane tipp 3, seega vastus on "YES".

Kolmandas päringus on tippude 1 ja 7 lähim ühine eellane tipp 4, seega vastus on "NO".

Neljandas päringus on tippude 4 ja 6 lähim ühine eellane tipp 4, seega vastus on "YES".

Seejärel saame väljastada, et juurtipp on 4, mis on ka õige vastus.

## Hindamine

1. (7 punkti):  $n \leq 9$ .
2. (10 punkti):  $n \leq 30$ .
3. (kuni 83 punkti):  $n \leq 500$ .

Kahes esimeses alamülesandes võid sa esitada kuni 1 000 päringut.

Kolmandas alamülesandes olgu  $k$  maksimaalne päringute arv, mille sinu programm ühes testis kulutas. Kui  $k \leq 9$ , saad sa 83 punkti. Vastasel juhul saad sa  $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$  punkti.

C++ kood, mis arvutab päringute arvust kolmanda alamülesande punktisumma:

```
((k <= 9) ? 83 : max(10, int(83 * (1 - log(k - 6.0) / 7))))
```