

Hol van a gyökér?

Ez egy interaktív feladat

Adott egy n csúcsból álló fa. A fa egy olyan gráf, ahol pontosan egy út van minden csúcspár között. **Szintén biztos, hogy létezik legalább egy olyan csúcs, amely legalább 3 csúccsal van közvetlenül összekötve.** A csúcsok egyike a fa gyökere, és az a feladatod, hogy megtaláld. Ennek érdekében a következő formájú kérdéseket teheted fel:

- Csúcsok egy adott a_1, a_2, \dots, a_m halmazára megkérdezheted, hogy legalacsonyabban lévő közös ősök e halmazban van-e.

Egy S csúcshalmaz egy v csúcsát a halmaz közös ősének nevezzük, ha az S -beli csúcsokból a gyökérbe vezető összes út átmegy v -n. Egy S csúcshalmaz legalacsonyabban lévő közös őse (lowest common ancestor, LCA) alatt S -nek azt a közös őst értjük, mely a gyökértől a legtávolabb helyezkedik el.

Interakció

Az interakció első lépéseként olvass be egy n egész számot ($4 \leq n \leq 500$) - a csúcsok számát.

Ezt követően olvass be $n - 1$ sort. Az i . sorban található két egész szám, a_i és b_i ($1 \leq a_i, b_i \leq n$) azt jelenti, hogy a fában a_i és b_i csúcsok között egy él fut.

Biztosan tudjuk, hogy ez az $n - 1$ él fát alkot, és hogy létezik legalább egy olyan csúcs, amely legalább 3 csúccsal van közvetlenül összekötve.

Egy kérdés feltevéséhez, először írd ki egy "?"-et, majd egy m egész számot, és végül m különböző a_1, a_2, \dots, a_m egész számot ($1 \leq m \leq n$, $1 \leq a_i \leq n$, minden a_i különböző), a csúcsokat, amelyekre le akarod ellenőrizni, hogy az LCA közöttük van-e.

Válaszként a válaszadó program "YES"-t ír ki, ha az LCA-juk egyike az a_1, a_2, \dots, a_m csúcsoknak, és "NO"-t egyébként.

Legfeljebb 1000 kérdést tehetsz fel, de a pontszámod függ a feltett kérdések számától. A válasz kiírása nem számít kérdésnek. A részletekhez nézd át a pontozásról szóló részt

Mikor meghatároztad a gyökeret, írd ki egy "!"-et, majd a gyökeret azonosító v egész számot ($1 \leq v \leq n$). Ezt követően állítsd le a programot.

Egy kérdés kiírása után ne felejts egy új sor karaktert írni és a kimenetet üríteni! Ez utóbbit a következőképpen tudod megtenni:

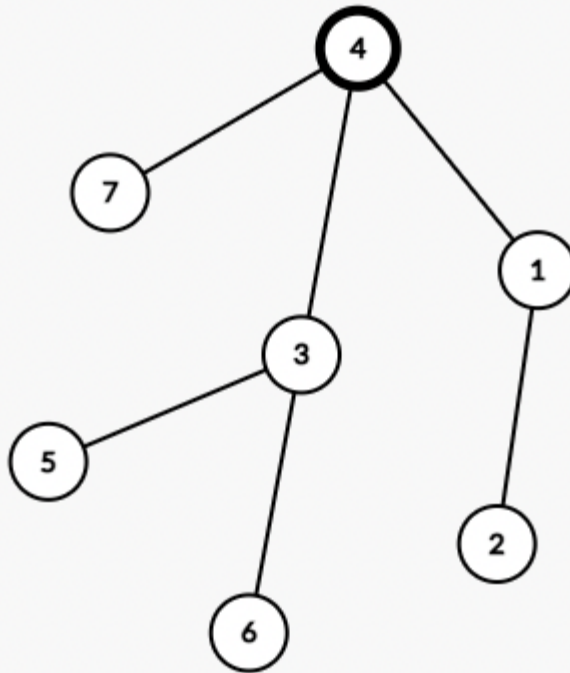
- `fflush(stdout)` vagy `cout.flush()` C++-ban;
- `stdout.flush()` Pythonban;

Mindegyik teszt esetén a fa és a gyökere rögzített az interakciók elkezdése előtt. Más szóval, a válaszadó program nem adaptív.

Példa

```
Input:
7
4 1
1 2
4 3
3 5
3 6
4 7
Output:
? 2 5 6
Input:
NO
Output:
? 3 6 3 5
Input:
YES
Output:
? 2 1 7
Input:
NO
Output:
? 2 4 6
Input:
YES
Output:
! 4
```

Magyarázat



Az ismeretlen gyökér a 4. csúcs.

Az első kérdésben az 5. és 6. csúcs LCA-ja a 3. csúcs, ami nincs közöttük, szóval a válasz "NO"

A második kérdésben a 3., 5. és 6. csúcs LCA-ja a 3. csúcs, tehát a válasz "YES"

A harmadik kérdésben az 1. és 7. csúcs LCA-ja a 4. csúcs, tehát a válasz "NO"

A negyedik kérdésben a 4. és 6. csúcs LCA-ja a 4. csúcs, tehát a válasz "YES"

Ezt követően ki tudjuk találni, hogy a gyökér a 4. csúcs, ami a helyes válasz.

Pontozás

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

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

3. (legfeljebb 83 pont): $n \leq 500$

Az első és második részfeladatban legfeljebb 1000 kérdést tehetsz fel.

A harmadik részfeladatban legyen k a kérdések maximális száma, amit feltettél bármelyik tesztben. Ha $k \leq 9$, akkor 83 pontot kapsz. Egyébként $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$ pontot kapsz.

C++ kód, ami kiszámítja a pontszámot a harmadik részfeladatban:

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