

Къде е коренът?

Това е една интерактивна задача

Дадено е дърво с n върха. Дървото е граф, така че има точно един прост път между всяка двойка върхове. **Също така е гарантирано, че поне един връх е директно свързан с ребро най-малко с 3 върха.** Един от върховете е коренът и вашата задача е да го намерите. За да направите това, имате право да задавате въпроси от вида:

- За даден списък a_1, a_2, \dots, a_m от върхове, проверете дали техният най-близък общ предшественик е в този списък.

Връх v е общ предшественик на списък S от върхове, ако пътищата от всички върхове в S до корена минават през v . Най-близкият общ предшественик (LCA) на списъка S от върхове е общият предшественик на S , който е най-отдалечен от корена.

Взаимодействие

Започнете взаимодействието, като прочетете едно цяло число n ($4 \leq n \leq 500$) - броят на върховете.

След това прочетете следващите $n - 1$ реда. i -тият ред ще съдържа две цели числа a_i, b_i ($1 \leq a_i, b_i \leq n$), показващи, че има ребро между възлите a_i, b_i в дървото.

Гарантирано е, че тези $n - 1$ ребра образуват дърво и че има точно един връх, директно свързан с ребро с най-малко 3 върха.

За да зададете въпрос, първо изведете "?", след това цялото число m и след това m различни цели числа a_1, a_2, \dots, a_m ($1 \leq m \leq n, 1 \leq a_i \leq n$, всички a_i са различни) - върхове, за които искате да проверите дали техният LCA е сред тях.

Като отговор интеракторът ще изведе "YES", ако техният LCA е един от a_1, a_2, \dots, a_m , и "NO" в противен случай.

Можете да зададете най-много 1000 въпроса, но ще получите различен брой точки в зависимост от това колко въпроса зададете. Извеждането на отговора не се счита за въпрос. Моля, погледнете раздела за точкуване за подробности.

Когато идентифицирате корена, изведете символа "!" и след това едно цяло число v ($1 \leq v \leq n$) - коренът. След това прекратете програмата си.

След като отпечатате въпрос, не забравяйте да изведете края на реда и да изчистите изхода. За да направите това, използвайте:

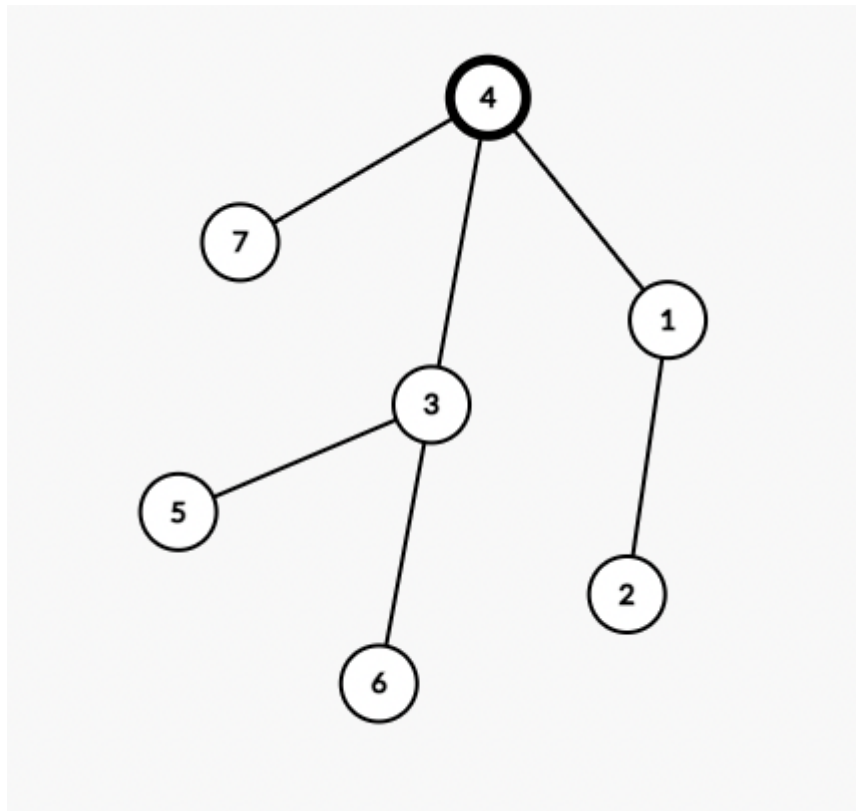
- `fflush(stdout)` или `cout.flush()` в C++;
- `stdout.flush()` в Python;

Гарантирано е, че за всеки тестов случай дървото и неговият корен са фиксирани преди началото на взаимодействието. С други думи, **интеракторът не е адаптивен**.

Пример

```
Вход
7
4 1
1 2
4 3
3 5
3 6
4 7
Изход:
? 2 5 6
Вход:
NO
Изход:
? 3 6 3 5
Вход:
YES
Изход:
? 2 1 7
Вход:
NO
Изход:
? 2 4 6
Вход:
YES
Изход:
! 4
```

Забележка



Скритият връх е 4.

В първия въпрос LCA на върхове 5 и 6 е връх 3, който не е сред върховете 5 и 6, така че отговорът е "NO".

Във втория въпрос LCA на върхове 3, 5 и 6 е връх 3, така че отговорът е "YES".

В третия въпрос LCA на върхове 1 и 7 е връх 4, така че отговорът е "NO".

В четвъртия въпрос LCA на върхове 4 и 6 е връх 4, така че отговорът е "YES".

След това можем да предположим, че коренът е връх 4, което е правилният отговор.

Оценяване

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

В първите две подзадачи можете да задавате най-много 1000 въпроса.

В третата подзадача, нека k е максималния брой въпроси във всеки тест. Ако $k \leq 9$, ще получите 83 точки. В противен случай ще получите $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$ точки.

C++ код, който изчислява броя на точките:

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