

## სად არის ფესვი?

ეს ამოცანა არის ინტერაქტიული.

მოცემული გაქვთ ხე  $n$  რაოდენობის წვეროთი. ხე არის გრაფი, რომელშიც თითოეულ ორ წვეროს შორის ზუსტად ერთი მარტივი გზა მოიძებნება. **გარანტირებულია, რომ მოცემულ ხეში ერთ-ერთი წვერო მაინც არის უშუალოდ დაკავშირებული მინიმუმ 3 წვეროსთან.** ხის ერთ-ერთი წვერო არის ფესვი (root) და თქვენ გევალებათ იპოვოთ ეს წვერო.

ამის გასაკეთებლად შეგიძლიათ დასვათ შემდეგი ფორმის შეკითხვები:

- მოცემული  $a_1, a_2, \dots, a_m$  წვეროების სიმრავლისათვის შეამოწმეთ მისი უდაბლესი საერთო წინაპარი (LCA) არის თუ არა ამავე სიმრავლეში.

შეგახსენებთ, რომ წვეროების სიმრავლის უდაბლესი საერთო წინაპარი (LCA) არის წვერო, რომელიც მდებარეობს ფესვიდან სიმრავლის თითოეულ წვერომდე მარტივ გზაზე და ყველაზე მეტად არის დაშორებული ფესვიდან.

წვერო  $v$  არის საერთო წინაპარი წვეროების სიმრავლე  $S$ -სთვის თუ გზები  $S$ -ის თითოეული წვეროდან ფესვამდე გადის  $v$ -ზე. უდაბლესი საერთო წინაპარი (LCA) წვეროების სიმრავლე  $S$ -სთვის არის  $S$ -ის საერთო წინაპარი, რომელიც ყველაზე მეტად არის დაშორებული ფესვიდან.

## ინტერაქცია

ინტერაქციის დასაწყისში უნდა წაიკითხოთ ერთი მთელი რიცხვი  $n$  ( $4 \leq n \leq 500$ ) - ხეში წვეროების რაოდენობა.

შემდეგ კითხულობთ  $n - 1$  რაოდენობის სტრიქონს.  $i$ -ური სტრიქონი შეიცავს ორ მთელ რიცხვს  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ), რომლებიც აღნიშნავენ, რომ  $a_i$  და  $b_i$  წვეროები დაკავშირებული არიან წიბოთი.

გარანტირებულია, რომ ეს  $n - 1$  რაოდენობის წიბო ქმნის ხეს და ამ ხის ერთი წვერო მაინც უშუალოდ დაკავშირებული მინიმუმ 3 წვეროსთან.

შეკითხვის დასასმელად ჯერ უნდა გამოიტანოთ "?", შემდეგ მთელი რიცხვი  $m$  და შემდეგ  $m$  ცალი განსხვავებული მთელი რიცხვი  $a_1, a_2, \dots, a_m$  ( $1 \leq m \leq n$ ,  $1 \leq a_i \leq n$  და ყველა  $a_i$  არის განსხვავებული) - წვეროთა სიმრავლე, რომელზეც გაიგებთ არის თუ არა მათი LCA ამავე სიმრავლეში.

პასუხად პროგრამა გამოიტანს "YES", თუ მათი LCA არის ერთ-ერთი  $a_1, a_2, \dots, a_m$ -ში, წინააღმდეგ შემთხვევაში გამოიტანს "NO".

შეგიძლიათ მაქსიმუმ 1000 შეკითხვის დასმა. პასუხის გამოტანა არ ითვლება შეკითხვად.

პასუხის გამოსატანად უნდა გამოიტანოთ სიმბოლო "!" და შემდეგ ერთი მთელი რიცხვი  $v$  ( $1 \leq v \leq n$ ) - ფესვი და დაასრულოთ პროგრამა.

თითოეული შეკითხვის და პასუხის გამოტანის შემდეგ არ დაგავიწყდეთ გამოტანილი მონაცემების ფლაშირება. ამისათვის გამოიყენეთ:

- `fflush(stdout)` ან `cout.flush()` C++-ში;
- `stdout.flush()` Python-ში;

გარანტირებულია, რომ თითოეული ტესტისთვის ხე და მისი ფესვი დაფიქსირებულია ინტერაქციის დაწყებამდე. სხვა სიტყვებით, **ინტერაქტორი არ არის ადაპტიური**.

## მაგალითი

შეტანა:

7

4 1

1 2

4 3

3 5

3 6

4 7

გამოტანა:

? 2 5 6

შეტანა:

NO

გამოტანა:

? 3 6 3 5

შეტანა:

YES

გამოტანა:

? 2 1 7

შეტანა:

NO

გამოტანა:

? 2 4 6

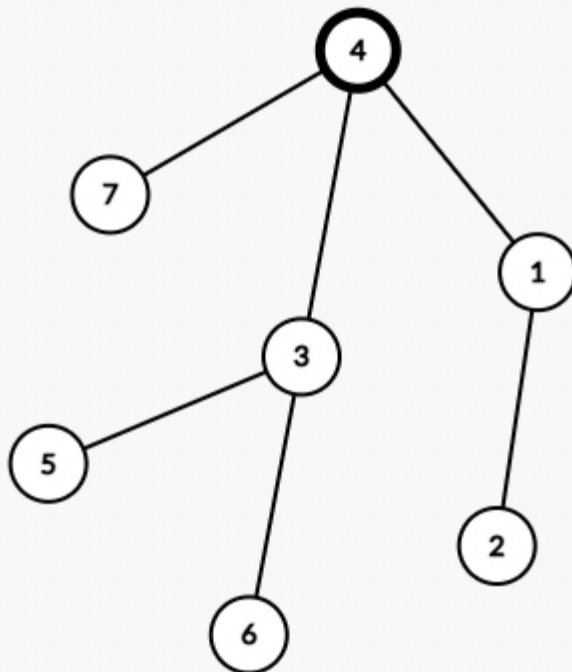
შეტანა:

YES

გამოტანა:

! 4

# განმარტება



დამალული ფესვი არის წვერო ნომერი 4.

პირველ შეკითხვაში 5 და 6 წვეროების LCA არის წვერო 3, რომელიც განსხვავებულია 5 და 6-სგან, ამიტომ პასუხი არის "NO".

მეორე შეკითხვაში 3, 5 და 6 წვეროების LCA არის წვერო 3, ამიტომ პასუხი არის "YES".

მესამე შეკითხვაში 1 და 7 წვეროების LCA არის წვერო 4, ამიტომ პასუხი არის "NO".

მეოთხე შეკითხვაში 4 და 6 წვეროების LCA არის წვერო 4, ამიტომ პასუხი არის "YES".

ამის შემდეგ უკვე შეგვიძლია გამოვიცნოთ, რომ ფესვი არის წვერო 4.

## შეფასება

1. (7 ქულა):  $n \leq 9$ ;
2. (10 ქულა):  $n \leq 30$ ;
3. (83 ქულამდე):  $n \leq 500$ .

პირველ და მეორე ქვეამოცანებში შეგიძლიათ დასვათ მაქსიმუმ 1000 შეკითხვა.

მესამე ქვეამოცანაში,  $k$ -თ აღვნიშნოთ ერთ ტესტზე თქვენს მიერ დასმული კითხვების მაქსიმალური რაოდენობა. თუ  $k \leq 9$ , თქვენ მიიღებთ 83 ქულას. წინააღმდეგ შემთხვევაში, თქვენ მიიღებთ  $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$  ქულას.

C++ კოდი, რომელიც დაითვლის ქულების რაოდენობას მესამე ქვეამოცანაში:

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