

Kök Nerede?

Bu iletişimli (interaktif) bir problemdir

Size n düğümlü bir ağaç veriliyor. Her düğüm ikilisi arasında tam olarak bir yolun bulunduğu çizgelere ağaç denir. **Düğümlerden en az bir tanesinin en az 3 komşusunun olduğu size garanti ediliyor.** Ağacın düğümlerinden bir tanesi kök olup, bu soruda sizden onu bulmanız isteniyor.

Aşağıdaki formatta sorgular yapmanıza izin veriliyor:

- Verilen bir a_1, a_2, \dots, a_m düğüm kümesi için, bu kümedeki düğümlerin en küçük ortak atasının (lowest common ancestor) kümede olup olmadığı söyle.

Ağacın kök düğümünü tahmin etmeniz gerekmektedir.

Bir v düğümü, verilen bir S düğüm kümesindeki her bir düğümden kök düğümüne olan yol üzerinde ise v düğümüne S kümesinin bir ortak atası denir. S kümesinin ortak atalarından düğümüne en uzak olanına S kümesinin en küçük ortak atası denir.

İletişim (Interaction)

İletişimi tek bir n ($4 \leq n \leq 500$) tamsayısı okuyarak başlatın. n ile ağaçtaki düğüm sayısı gösterilmektedir.

Sonra sıradaki $n - 1$ satısı okuyun. i . satır $1 \leq a_i, b_i \leq n$ şartını sağlayan a_i ve b_i tamsayılarından oluşmaktadır. Bu satır ağaçtaki a_i ve b_i düğümleri arasında bir kenar (edge) olduğunu göstermektedir.

Verilen $n - 1$ kenarın bir ağaç olduğu ve en az bir düğümün komşu sayısının (derecesinin) en az 3 olduğu size garanti edilmektedir.

Bir soru sormak için ilk olarak "?" bastırın, sonra bir m ($1 \leq m \leq n$) tamsayısı bastırın, ve son olarak en küçük ortak atasının içlerinden biri olup olmadığını kontrol etmek istediğiniz m tane birbirinden farklı a_1, a_2, \dots, a_m düğümün numarasını bastırın. $1 \leq a_i \leq n$ şartını sağlamaktadır ve bütün a_i değerleri birbirinden farklıdır.

Sorunuza cevap olarak, eğer sorudaki a_1, a_2, \dots, a_m düğümlerinin en küçük ortak atası içlerinden biri ise program "YES" bastıracaktır, aksi takdirde "NO" bastıracaktır.

En fazla 1000 soru sorabilirsiniz. Cevabı bastırmak soru sormadan sayılmamaktadır.

Kök düğümü belirlediğinizde, ilk olarak "!" sembolünü bastırın, sonrasında kök düğümün numarası olan v ($1 \leq v \leq n$) tamsayısını bastırın. Daha sonra programınızı sonlandırın.

Bir soru yazdırdıktan sonra satır sonunu (end of line) yazmayı ve çıktıyı flush etmeyi unutmayın. Bunu yapmak için aşağıdaki fonksiyonları kullanınız:

- C++'da: `fflush(stdout)` veya `cout.flush();`
- Python'da: `stdout.flush()`.

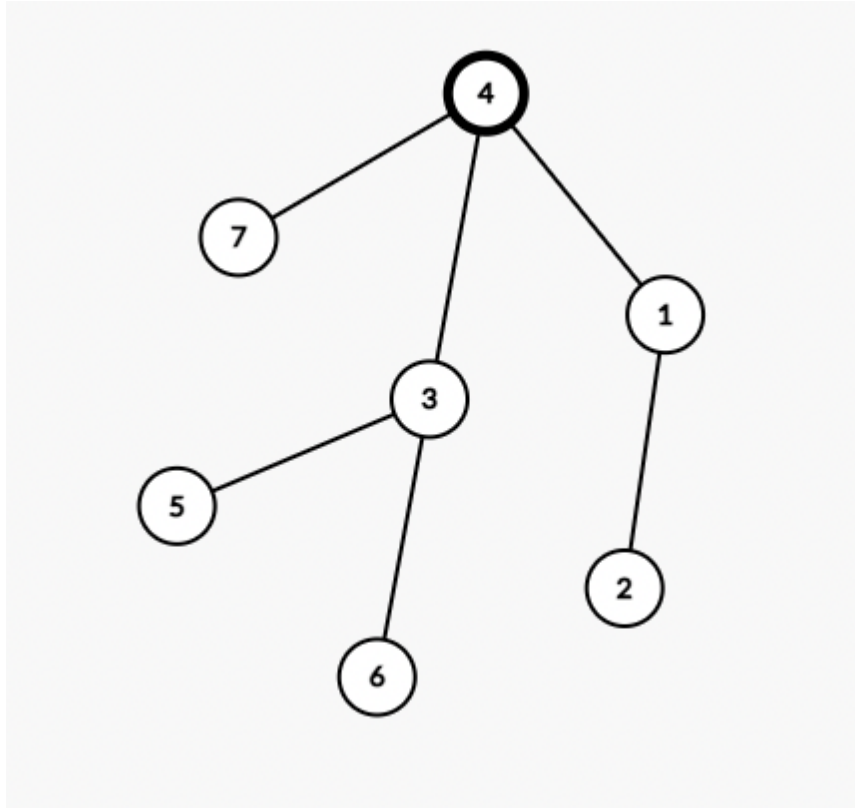
Her test durumunda ağacın ve kökünün iletişimden önce sabitlendiği garanti edilmektedir. Yani, **program adaptif değildir**.

Her sorudan sonra flush etmeyi unutmayın.

Örnek

```
Girdi:
7
4 1
1 2
4 3
3 5
3 6
4 7
Çıktı:
? 2 5 6
Girdi:
NO
Çıktı:
? 3 6 3 5
Girdi:
YES
Çıktı:
? 2 1 7
Girdi:
NO
Çıktı:
? 2 4 6
Girdi:
YES
Çıktı:
! 4
```

Not



Bu örnekte gizli kök 4 numaralı düğümdür.

İlk soruda: 5 ve 6 numaralı düğümlerin en küçük ortak arası 3 numaralı düğümdür ve dolayısıyla 5 ve 6 numaralı düğümlerden birisi değildir. Bu yüzden cevap "NO".

İkinci soruda: 3, 5 ve 6 numaralı düğümlerin en küçük ortak arası 3 numaralı düğümdür. Bu yüzden cevap "YES".

Üçüncü soruda: 1 ve 7 numaralı düğümlerin en küçük ortak arası 4 numaralı düğümdür. Bu yüzden cevap "NO".

Dördüncü soruda: 4 ve 6 numaralı düğümlerin en küçük ortak arası 4 numaralı düğümdür. Bu yüzden cevap "YES".

Bu sorular sonrasında, doğru cevap olan 4 numaralı düğümün kök düğüm olduğunu tahmin edebiliriz.

Puanlama

k tamsayısı, bir altgörevde kullandığınız en fazla soru sayısını gösterebilir. En fazla 1000 sorgu yapabileceğiniz için $k \leq 1000$ 'dir.

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

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

3. (en fazla 83 puan): $n \leq 500$

Üçüncü altgörevde, $k \leq 9$ ise 83 puan alırsınız. Aksi taktirde, $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$ puan alırsınız.

Bu yüzden tam puan almak için, çözümünüz bu altgörevdeki her test durumu için en fazla 9 sorgu yapmalıdır.

Bu altgörevden alacağınız puanı hesaplayan C++ kodu aşağıdadır:

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