

Where Is the Root?

Este é um problema interativo

É-te dada uma árvore com n vértices. A árvore é um grafo onde existe exatamente um único caminho simples entre cada par de vértices. **É também garantido que pelo um vértice está diretamente ligado a pelo menos 3 outros vértices.** Um dos vértices é a raiz, e a tua tarefa é descobri-lo. Para atingir este objetivo, é permitido que faças perguntas no seguinte formato:

- Para um dado conjunto a_1, a_2, \dots, a_m de vértices, verifica se o seu ancestral comum mais próximo (*lowest common ancestor*) está no conjunto.

Um vértice v é um ancestral comum de um conjunto S de vértices se os caminhos de todos os vértices de S para a raiz passam por v . O ancestral comum mais próximo (LCA) de um conjunto S de vértices é o ancestral comum de S que está o mais longe possível da raiz.

Interação

Começa a interação lendo um único inteiro n ($4 \leq n \leq 500$) - o número de vértices.

Depois lê as $n - 1$ linhas seguintes. A i -ésima linha irá conter dois inteiros a_i, b_i ($1 \leq a_i, b_i \leq n$), indicando que existe uma ligação entre os nós a_i, b_i na árvore.

É garantido que estas $n - 1$ ligações formam uma árvore e que pelo menos um vértice está diretamente ligado a pelo menos 3 outros vértices.

Para fazer uma pergunta escreve primeiro "?", e depois o inteiro m , seguido de m inteiros distintos a_1, a_2, \dots, a_m ($1 \leq m \leq n$, $1 \leq a_i \leq n$, todos os a_i são distintos) - os vértices para os quais queres verificar se o seu LCA está entre eles.

Como resposta, o programa irá escrever "YES" se o seu LCA é um dos elementos a_1, a_2, \dots, a_m e "NO" caso contrário.

Podes fazer um máximo de 1000 perguntas, mas irás obter um número diferente de pontos dependendo de quantas perguntas fizeres. Escrever a resposta não conta como pergunta. Por favor verifica a secção da pontuação para mais detalhes.

Quando tiveres identificado a raiz, escreve o símbolo "!" e depois um inteiro v ($1 \leq v \leq n$) - a raiz. Depois, termina o teu programa.

Depois de escrever uma pergunta, não te esqueças de escrever uma mudança de linha e fazer *flush* ao output. Para isso usa:

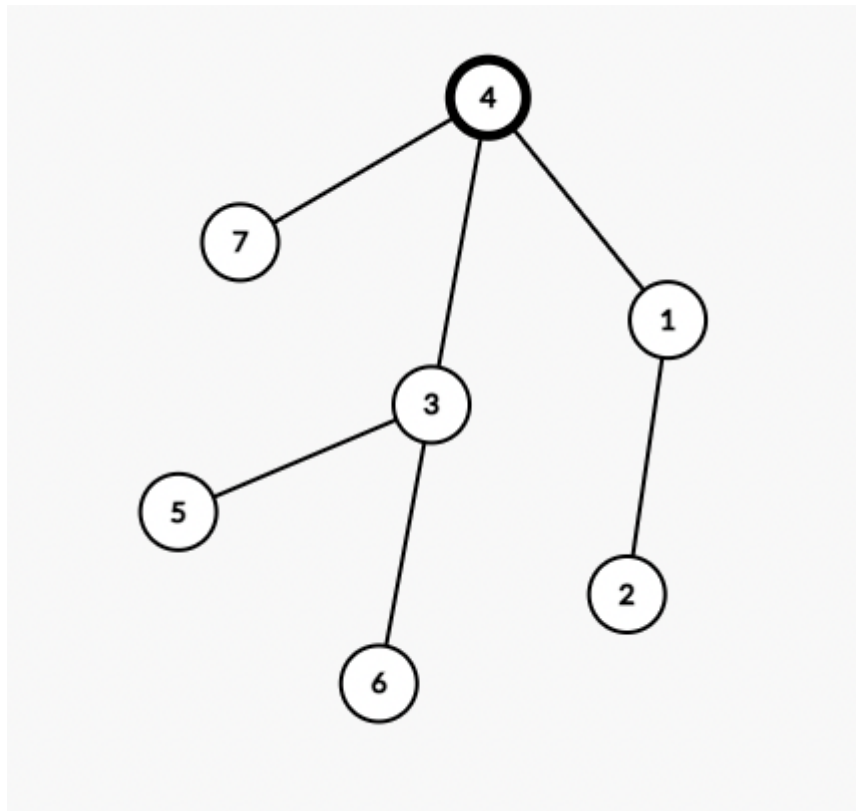
- `fflush(stdout)` ou `cout.flush()` em C++;
- `stdout.flush()` em Python;

É garantido que para cada caso de teste, a árvore e a sua raíz estão fixas antes do início da interação. Por outras palavras, **o programa que interage contigo não é adaptativo.**

Exemplo

```
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
```

Nota



A raiz escondido é o vértice 4.

Na primeira pergunta, o LCA dos vértices 5 e 6 é o vértice 3, que não está entre os vértices 5 e 6 e por isso a resposta é "NO".

Na segunda pergunta, o LCA dos vértices 3, 5 e 6 é o vértice 3 e por isso a resposta é "YES".

Na terceira pergunta, o LCA dos vértices 1 e 7 é o vértice 4 e por isso a resposta é "NO".

Na quarta pergunta, o LCA dos vértices 4 e 6 é o vértice 4 e por isso a resposta é "YES".

Depois disto, podemos adivinhar que a raiz é o vértice 4, que é a resposta correta.

Pontuação

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

Na primeiro e na segunda subtarefas podes fazer um máximo de 1000 perguntas.

Na terceira subtarefa, seja k o número máximo de perguntas que fazes num qualquer teste. Se $k \leq 9$, terás 83 pontos. De outro modo, irás obter $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$ pontos.

Código C++ que imprime o número de pontos para a terceira subtarefa:

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