

Tree Search

Se dă un arbore binar cu rădăcină care conține N vârfuri. Vârfurile sunt numerotate de la 1 la N , iar vârful 1 este rădăcina arborelui. Fiecare dintre celelalte vârfuri are un singur părinte. Deoarece arborele este binar, fiecare vârf poate fi părintele la cel mult alte două vârfuri.

Unul dintre vârfuri este special. Încearcă să îl ghicești. Pentru aceasta poți adresa întrebări în felul următor: „Este vârful special în subarborele vârfului x ”? Un vârf y se află în subarborele vârfului x dacă și numai dacă cel mai scurt drum între y și vârful 1 trece prin vârful x . Țineți cont de faptul că vârful x se află în subarborele vârfului x .

Aveți dreptul să adresați această întrebare de cel mult 35 de ori. După aceasta trebuie să raportați vârful ghicit.

Detalii de implementare

Trebuie să implementați următoarea funcție:

```
int solve(int N, std::vector < int > p)
```

- N : numărul de vârfuri
- p conține exact $N - 1$ elemente care descriu arborele: vârful $p[i]$ (unde $1 \leq p[i] \leq i + 1$) este părintele vârfului $i + 2$ pentru fiecare $0 \leq i \leq N - 2$
- Nici un element în p nu apare mai mult de două ori
- Funcția trebuie să returneze numărul vârfului special
- Această funcție va fi apelată o singură dată

Funcția de mai sus poate face apeluri la următoarea funcție:

```
int ask(int x)
```

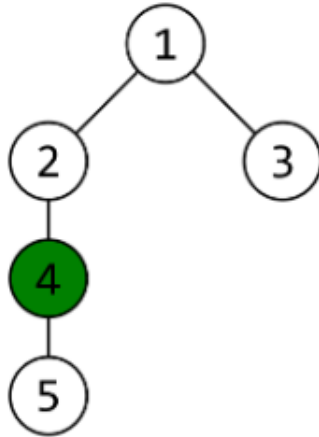
- x : numărul vârfului
- $1 \leq x \leq N$
- returnează 1 dacă vârful special este în subarborele x și 0 în caz contrar

Exemplu

Se consideră următorul apel:

```
solve(5, [1, 1, 2, 4])
```

Arborele constă din muchiile (1,2), (1,3), (2,4) și (4,5).



Programul tău face un apel

```
ask(4)
```

care returnează 1. După aceasta programul tău face un apel

```
ask(5)
```

care returnează 0.

Programul concluzionează că vârful 4 este special și returnează 4.

Restricții

- $2 \leq N \leq 100\,000$

Subtask-uri

1. (20 puncte) $N \leq 35$
2. (30 puncte) $p[i] = i + 1$ pentru fiecare $0 \leq i \leq N - 2$
3. (15 puncte) $p[i] = \lfloor i/2 \rfloor + 1$ pentru fiecare $0 \leq i \leq N - 2$
4. (35 puncte) Fără restricții adiționale.

Exemplu de Grader

Exemplu de Grader citește input-ul în următorul format:

- linia 1: N
- linia 2: $p[0], p[1], \dots, p[N - 2]$

Exemplu de Grader afișează fiecare întrebare în următorul format::

- linia 1: ? x

Exemplu de Grader citește fiecare răspuns în următorul format:

- linia 1: y

Exemplu de Grader afișează vârful ghicit în următorul format:

- linia 1: ! x