

Zoeken in de boom (Tree Search)

Je krijgt een binaire boom met N knopen en met een gegeven wortel. De knopen zijn genummerd van 1 tot en met N en de wortel knoop draagt het nummer 1. Alle andere knopen hebben één ouder in de boom. De boom is binair, in andere woorden: ieder knoop kan de ouder zijn van maximaal twee andere knopen.

Één van de knopen is speciaal. Je doel is om uit te zoeken welke het is. Je mag de volgende vraag stellen: "Is de speciale knoop in de deelboom van knoop x "? Een knoop y is in de deelboom van knoop x als en slechts als de kortste weg tussen y en 1 door knoop x gaat. Merk op dat knoop x ook in zijn eigen deelboom is.

Je mag maximaal 35 vragen stellen. Daarna moet je een antwoord teruggeven.

Implementatiedetails

Je moet de volgende functie implementeren:

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

- N : het aantal knopen
- p bestaat uit precies $N - 1$ elementen die de boom beschrijven: knoop $p[i]$ (met $1 \leq p[i] \leq i + 1$) is de ouder van knoop $i + 2$ voor alle i zodat $0 \leq i \leq N - 2$
- Geen element in p komt meer dan twee keer voor
- Deze functie moet het nummer van de speciale knoop teruggeven
- Deze functie wordt precies eens opgeroepen

De bovenstaande functie mag de onderstaande functie oproepen:

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

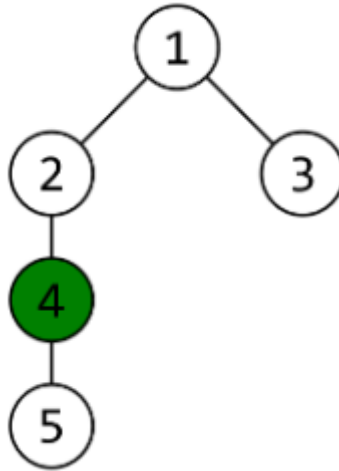
- x : het nummer van de knoop
- $1 \leq x \leq N$
- geeft 1 terug als de speciale knoop in de deelboom van x zit, anders geeft 0 terug

Voorbeeld

Gegeven de volgende oproep:

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

De boom bestaat uit takken (1,2), (1,3), (2,4) en (4,5).



Je programma roep dit op:

```
ask(4)
```

die 1 teruggeeft. Daarna roept je programma dit op:

```
ask(5)
```

die 0 teruggeeft.

Je programma concludeert dat knoop 4 de speciale knoop is en geeft dus 4 terug.

Constraints

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

Subtasks

1. (20 punten) $N \leq 35$
2. (30 punten) $p[i] = i + 1$ voor elke i zodat $0 \leq i \leq N - 2$
3. (15 punten) $p[i] = \lfloor i/2 \rfloor + 1$ voor elke i zodat $0 \leq i \leq N - 2$
4. (35 punten) Geen bijkomende beperkingen.

Sample Grader

De sample grader leest de invoer in het volgende format:

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

De sample grader print elke vraag in het volgende format:

- regel 1: ? x

De sample grader leest elke antwoord in het volgende format:

- regel 1: y

De sample grader print de door jou gevonden speciale knoop in het volgende format:

- regel 1: ! x