

Przeszukiwanie drzewa

Dane jest ukorzenione drzewo binarne zawierające N wierzchołków. Wierzchołki numerowane są kolejnymi liczbami od 1 do N , korzeń drzewa to wierzchołek numer 1. Każdy z pozostałych wierzchołków ma dokładnie jednego rodzica w drzewie. Drzewo jest binarne, tzn. każdy wierzchołek może być rodzicem co najwyżej dwóch innych wierzchołków.

Jeden z wierzchołków jest specjalny. Twoim celem jest go odgadnąć. Możesz zadawać pytania następującej postaci: "Czy specjalny wierzchołek znajduje się w poddrzewie wierzchołka x ". Wierzchołek y jest w poddrzewie wierzchołka x wtedy i tylko wtedy, gdy najkrótsza ścieżka pomiędzy y a 1 przechodzi przez wierzchołek x . Możesz zadać takie pytanie co najwyżej 35 razy. Potem musisz podać swoją odpowiedź.

Szczegóły implementacyjne

Należy zaimplementować poniższą funkcję:

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

- N : liczba wierzchołków
- p zawiera dokładnie $N - 1$ elementów opisujących drzewo: wierzchołek $p[i]$ (gdzie $1 \leq p[i] \leq i + 1$) jest rodzicem wierzchołka $i + 2$ dla każdego $0 \leq i \leq N - 2$
- Żaden element w p nie występuje więcej niż raz
- Funkcja powinna zwrócić numer specjalnego wierzchołka
- Ta funkcja będzie wywołana tylko raz

Powyższa funkcja może wywoływać poniższą funkcję:

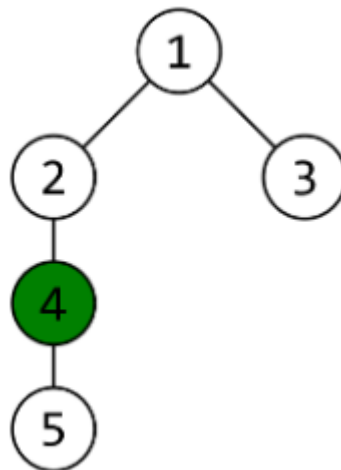
```
int ask(int x)
```

- x : numer wierzchołka
- $1 \leq x \leq N$
- zwraca 1 jeżeli specjalny wierzchołek jest w poddrzewie x i 0 w przeciwnym przypadku

Przykład

Rozważmy poniższe wywołanie:

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



Drzewo zawiera krawędzie (1,2), (1,3), (2,4) i (4,5).

Twój program wywołał

```
ask(4)
```

które zwróciło 1. Następnie Twój program wywołał

```
ask(5)
```

które zwróciło 0.

Twój program wywnioskował, że wierzchołek 4 jest specjalny i zwrócił 4.

Ograniczenia

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

Podzadania

1. (20 punktów) $N \leq 35$
2. (30 punktów) $p[i] = i + 1$ dla każdego $0 \leq i \leq N - 2$
3. (15 punktów) $p[i] = \lfloor i/2 \rfloor + 1$ dla każdego $0 \leq i \leq N - 2$
4. (35 punktów) Brak dodatkowych ograniczeń.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka wczytuje wejście w następującej formie:

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

Przykładowa sprawdzaczka wypisuje każde pytanie w następującym formacie:

- wiersz 1: ? x

Przykładowa sprawdzaczka wczytuje każdą odpowiedź w następującym formacie:

- wiersz 1: y

Przykładowa sprawdzaczka wypisuje odpowiedź w następującym formacie:

- wiersz 1: ! x