

Пошук в дереві

Вам дано кореневе бінарне дерево, що складається з N вершин. Вершини пронумеровані від 1 до N , коренем є вершина з номером 1. Всі інші вершини мають одного батька в дереві. Дерево є бінарним, тобто кожна вершина може бути батьком щонайбільше двох інших вершин.

Одна з вершин є особливою. Ви намагаєтеся її вгадати. Ви можете ставити питання такого типу: «Чи правда, що особлива вершина знаходиться в піддереві вершини x »? Вершина y належить піддереву вершини x тоді і лише тоді, коли найкоротший шлях між y і 1 проходить через вершину x . Зауважте, що вершина x також належить своєму власному піддереву.

Вам дозволено поставити таке запитання не більше 35 разів. Після цього ви повинні повідомити свою здогадку.

Деталі реалізації

Ви повинні реалізувати таку функцію:

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

- N : кількість вершин
- p складається з рівно $N - 1$ елементів, які описують дерево: вершина $p[i]$ (де $1 \leq p[i] \leq i + 1$) є батьком вершини $i + 2$ для кожного $0 \leq i \leq N - 2$
- В p немає елементів, що зустрічаються більше ніж 2 рази.
- Ця функція повинна повернути номер особливої вершини.
- Ця функція викликається рівно один раз

Наведена вище функція може викликати наступну функцію:

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

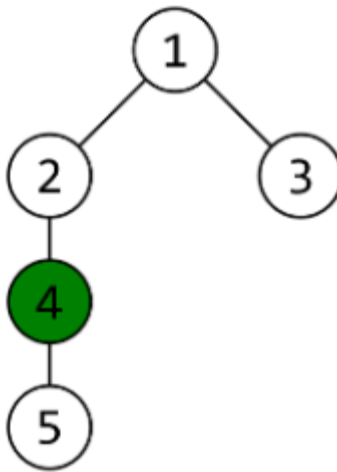
- x : номер вершини
- $1 \leq x \leq N$
- повертає 1 якщо особлива вершина знаходиться в піддереві x та 0 в іншому випадку

Приклад

Розглянемо такий виклик функції:

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

Дерево складається з ребер (1,2), (1,3), (2,4) та (4,5).



Ваша програма зробила виклик

```
ask(4)
```

який повернув 1. Після цього ваша програма зробила виклик

```
ask(5)
```

який повернув 0.

Ваша програма дійшла висновку, що вершина 4 є особливою, і повернула 4.

Обмеження

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

Підзадачі

1. (20 балів) $N \leq 35$
2. (30 балів) $p[i] = i + 1$ для кожного $0 \leq i \leq N - 2$
3. (15 балів) $p[i] = \lfloor i/2 \rfloor + 1$ для кожного $0 \leq i \leq N - 2$
4. (35 балів) Без додаткових обмежень.

Приклад градера

Приклад градера зчитує вхідні дані в такому форматі:

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

Приклад градера виводить кожне запитання в такому форматі:

- рядок 1: $? x$

Приклад градера зчитує кожну відповідь в такому форматі:

- рядок 1: y

Приклад градера виводить здогадку в такому форматі:

- рядок 1: $! x$