# C. Приснится же такое...

✓ Полное решение </> OK Python 3.12.3

## Исходный код

```python
from collections import deque


class Node:
    def __init__(self, value, parent=None, left=None, right=None) -> None:
        self.value = value
        self.left = left
        self.right = right
        self.parent = parent


def swap_with_left_child(p: Node, v: Node, node_map: dict[int, Node]):
    if v.right:
        v.right.parent = p
    if p.right:
        p.right.parent = v

    p.value, v.value = v.value, p.value
    p.right, v.right = v.right, p.right
    node_map[v.value] = v
    node_map[p.value] = p


def swap_with_right_child(p: Node, v: Node, node_map: dict[int, Node]):
    if v.left:
        v.left.parent = p
    if p.left:
        p.left.parent = v

    p.value, v.value = v.value, p.value
    p.left, v.left = v.left, p.left
    node_map[v.value] = v
    node_map[p.value] = p


def swap(p: Node, v: Node, node_map: dict[int, Node]):
    if p.left == v:
        swap_with_left_child(p, v, node_map)
    else:
        swap_with_right_child(p, v, node_map)


def build_tree(n: int) -> tuple[Node, dict[int, Node]]:
    root = Node(1)
    queue = deque()
    queue.append(root)
    current_node_id = 2
    node_map = {1: root}

    while queue and current_node_id <= n:
        parent = queue.popleft()
```

```
        if current_node_id <= n:
            parent.left = Node(current_node_id, parent)
            queue.append(parent.left)
            node_map[current_node_id] = parent.left
            current_node_id += 1
        if current_node_id <= n:
            parent.right = Node(current_node_id, parent)
            queue.append(parent.right)
            node_map[current_node_id] = parent.right
            current_node_id += 1

    return root, node_map


def lvr(root: Node):
    if not root:
        return

    stack = []
    curr = root

    while curr or stack:
        while curr:
            stack.append(curr)
            curr = curr.left

        curr = stack.pop()
        print(curr.value, end=" ")
        curr = curr.right


def read_words(f, buffer_size=4096):
    buffer = ''

    while True:
        chunk = f.read(buffer_size)

        if not chunk:
            if buffer:
                yield from buffer.split()
            break

        buffer += chunk
        parts = buffer.split()

        if not buffer or buffer[-1].isspace():
            yield from parts
            buffer = ''
        else:
            yield from parts[:-1]
            buffer = parts[-1] if parts else ''


def main(filename: str) -> Node:
    with open(filename) as f:
        N = int(f.readline().split()[0])
        root, node_map = build_tree(N)

        for word in read_words(f):
            node_id = int(word)
            node = node_map[node_id]

            if node == root:
                continue

            swap(node.parent, node, node_map)

    return root


if __name__ == "__main__":
```

```
124     root = main("input.txt")
125     lvr(root)
126
```

Отличия от предыдущей посылки

| № | Вердикт | Ресурсы | Баллы |
|---|---|---|---|
| 1 | ok | 50ms / 4.64Mb | — |
| 2 | ok | 50ms / 4.64Mb | — |
| 3 | ok | 50ms / 4.64Mb | — |
| 4 | ok | 50ms / 4.64Mb | — |
| 5 | ok | 50ms / 4.64Mb | — |
| 6 | ok | 50ms / 4.64Mb | — |
| 7 | ok | 51ms / 4.64Mb | — |
| 8 | ok | 50ms / 4.64Mb | — |
| 9 | ok | 50ms / 4.64Mb | — |
| 10 | ok | 49ms / 4.64Mb | — |
| 11 | ok | 50ms / 4.64Mb | — |
| 12 | ok | 50ms / 4.64Mb | — |
| 13 | ok | 0.712s / 4.64Mb | — |
| 14 | ok | 0.668s / 4.64Mb | — |
| 15 | ok | 0.682s / 4.64Mb | — |
| 16 | ok | 0.686s / 4.64Mb | — |
| 17 | ok | 0.662s / 4.64Mb | — |
| 18 | ok | 0.662s / 4.64Mb | — |
| 19 | ok | 0.66s / 4.64Mb | — |
| 20 | ok | 0.678s / 4.64Mb | — |
| 21 | ok | 0.687s / 4.64Mb | — |
| 22 | ok | 0.629s / 4.64Mb | — |
| 23 | ok | 0.635s / 4.64Mb | — |
| 24 | ok | 0.631s / 4.64Mb | — |
| 25 | ok | 0.637s / 4.64Mb | — |
| 26 | ok | 0.633s / 4.64Mb | — |
| 27 | ok | 0.635s / 4.64Mb | — |
| 28 | ok | 50ms / 4.64Mb | — |
| 29 | ok | 0.501s / 4.64Mb | — |
| 30 | ok | 0.559s / 4.64Mb | — |

| № | Вердикт | Ресурсы | Баллы |
|---|---|---|---|
| 31 | ok | 0.594s / 4.64Mb | — |
| 32 | ok | 0.612s / 4.64Mb | — |
| 33 | ok | 0.618s / 4.64Mb | — |
| 34 | ok | 0.63s / 4.64Mb | — |
| 35 | ok | 0.635s / 4.64Mb | — |
| 36 | ok | 0.646s / 4.64Mb | — |
| 37 | ok | 0.662s / 4.64Mb | — |
| 38 | ok | 0.772s / 4.51Mb | — |
| 39 | ok | 0.762s / 4.64Mb | — |
| 40 | ok | 0.748s / 4.64Mb | — |
| 41 | ok | 0.747s / 4.64Mb | — |
| 42 | ok | 0.738s / 4.64Mb | — |
| 43 | ok | 0.736s / 4.64Mb | — |
| 44 | ok | 0.731s / 4.64Mb | — |
| 45 | ok | 0.725s / 4.64Mb | — |
| 46 | ok | 0.725s / 4.64Mb | — |
| 47 | ok | 0.72s / 4.64Mb | — |