

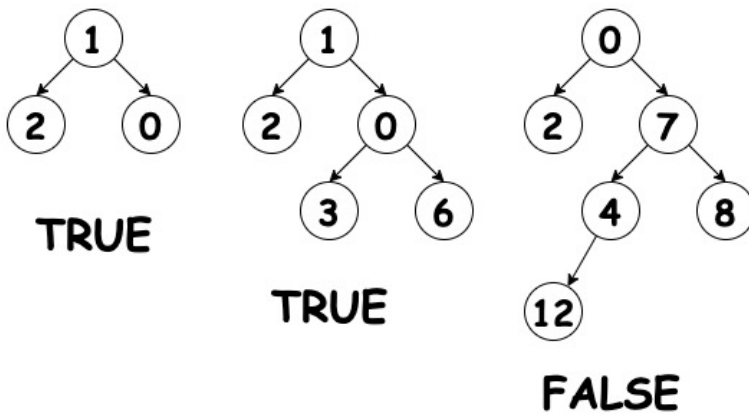
# Яндекс.Практикум, Python-разработчик: Спринт 15. Деревья.

## В. Сбалансированное дерево

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Гоше очень понравилось слушать рассказ Тимофея про деревья. Особенно часть про сбалансированные деревья. Он решил написать функцию, которая определяет, сбалансировано ли дерево.

Дерево считается сбалансированным, если левое и правое поддеревья каждого узла отличаются по высоте не больше, чем на один.



### Формат ввода

На вход функции подается корень бинарного дерева.

Python:

```
class Node:
    def __init__(self, value, left=None, right=None):
        self.value = value
        self.right = right
        self.left = left
```

Ваша функция должна иметь сигнатуру `solution(Node root) -> bool`.

C++:

```
struct Node{
    int value;
    const Node* left = nullptr;
    const Node* right = nullptr;
};
bool Solution(const Node* root);
```

Нужно подключить `solution_tree.h`

Go:

```
type TNode struct {
    value int
    left *TNode
    right *TNode
}
```

Ваша функция должна иметь сигнатуру `Solution(root *TNode) bool`.

JS:

```
class CNode {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}
```

Ваша функция должна иметь сигнатуру `Solution(root) :: CNode -> bool`.

Java:

Файл должен содержать `public class Solution` с `public static boolean treeSolution(Node head)`

```
class Node {
  int value;
  Node left;
  Node right;

  Node(int value) {
    this.value = value;
    right = null;
    left = null;
  }
}
```

## Формат вывода

Функция должна вернуть `True`, если дерево сбалансировано в соответствии с критерием из условия, иначе - `False`.

## Примечания

Решение нужно отправлять в виде файла с расширением соответствующем вашему языку программирования. Нужно выбирать компилятор `make`.

Для Java файл должен называться `Solution.java`

Для остальных языков программирования это имя использовать нельзя (имя `solution` тоже).