

# Яндекс. Тренировки по алгоритмам июнь 2021, занятие 8

## А. Высота дерева

	Все языки	Python 3.6
Ограничение времени	2 секунды	4 секунды
Ограничение памяти	64Mb	256Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Реализуйте бинарное дерево поиска для целых чисел. Программа получает на вход последовательность целых чисел и строит из них дерево. Элементы в деревья добавляются в соответствии с результатом поиска их места. Если элемент уже существует в дереве, добавлять его не надо. Балансировка дерева не производится.

### Формат ввода

На вход программа получает последовательность натуральных чисел. Последовательность завершается числом 0, которое означает конец ввода, и добавлять его в дерево не надо.

### Формат вывода

Выведите единственное число – высоту получившегося дерева.

### Пример

Ввод

7 3 2 1 9 5 4 6 8 0

Вывод

4

Язык

```
1 class Node:
2     def __init__(self, val=None, left=None, right=None):
3         self.val = val
4         self.left = left
5         self.right = right
6
7
8 class Tree:
9     def __init__(self):
10         self.root = None
11
12     def height(self):
13
14         def _traverse(node):
15             if node is None:
16                 return 0
17             left_height = _traverse(node.left)
18             right_height = _traverse(node.right)
19             return max(left_height, right_height) + 1
20
21         return _traverse(self.root)
22
23     def add(self, num):
24
25         def _add(node, num):
26             if node.val > num:
27                 if node.left is None:
28                     node.left = Node(num)
29                 else:
30                     _add(node.left, num)
31             elif node.val < num:
32                 if node.right is None:
33                     node.right = Node(num)
34                 else:
35                     _add(node.right, num)
36
37         if self.root is None:
38             self.root = Node(num)
```

Отправить

Следующая