

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и Структуры Данных»
Тема: Вычисление высоты дерева

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Иванов Д. В.

Санкт-Петербург

2022

Цель работы.

Реализовать программу, производящую вычисление и вывод высоты дерева, заданного как последовательность чисел, и тестирование к ней.

Задание.

Вычисление высоты дерева. Python.

На вход программе подается корневое дерево с вершинами $\{0, \dots, n-1\}$, заданное как последовательность $\text{parent}_0, \dots, \text{parent}_{n-1}$, где parent_i — родитель i -й вершины. Требуется вычислить и вывести высоту этого дерева.

Формат входа.

Первая строка содержит натуральное число n . Вторая строка содержит n целых чисел $\text{parent}_0, \dots, \text{parent}_{n-1}$. Для каждого $0 \leq i \leq n-1$, parent_i — родитель вершины i ; если $\text{parent}_i = -1$, то i является корнем. Гарантируется, что корень ровно один и что данная последовательность задаёт дерево.

Формат выхода.

Высота дерева.

Примечание: высотой дерева будем считать количество вершин в самом длинном пути от корня к листу.

Выполнение работы.

Make_tree:

Функция создает два массива *parents* и *leafs*, в одном хранится список родителей, поданный на вход, другой массив числа от 0 до длины первого массива. Второй массив будет хранить в себе «листья» дерева. Далее с помощью цикла определяется, присутствует ли число в списке родителей, и если да, то удаляется из второго списка. По итогу во втором массиве должны остаться только «листья».

Find_length(leafs, parents):

Функция принимает на вход полученные ранее массивы. С помощью цикла перебираются все «листья», и у каждого определяется длина до корня. Функция возвращает максимальную из вычисленных длин.

Main:

Вызывает *Make_tree* и в случае, если дерево не состоит из одного элемента, вызывает *Find_length(leafs, parents)* и выводится полученная длина, иначе выводится длина 1.

Вывод.

Были освоены навыки работы с деревьями и принципы тестирования. Разработана программа, вычисляющая и выводящая на экран высоту дерева, заданного с клавиатуры в виде последовательности чисел. Реализовано осуществлено тестирование.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 — результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	5 -1 0 4 0 3	4	Верно
2.	1 -1	1	Верно
3.	8 7 0 6 2 5 1 4 -1	8	Верно

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from modules.find_length import find_length
from modules.make_tree import make_tree

def main():
    trash = input()
    parents = [int(x) for x in input().split()]
    leafs, parents = make_tree(parents)
    if len(parents) == 1:
        print(1)
    else:
        print(find_length(leafs, parents))

if __name__ == '__main__':
    main()
```

Название файла: make_tree.py

```
def make_tree(parents):
    leafs = [i for i in range(0, len(parents))]
    for i in range(len(parents)):
        try:
            leafs.remove(parents[i])
        except:
            pass
    return leafs
```

Название файла: find_length.py

```
def find_length(leafs, parents):
    mx_len = 0
    for i in range(len(leafs)):
        cur_len = 1
```

```

    cur_parent = parents[leafs[i]]
    while cur_parent != -1:
        cur_parent = parents[cur_parent]
        cur_len += 1
    if cur_len > mx_len:
        mx_len = cur_len
    return mx_len

```

Название файла: tests.py

```

from modules.find_length import find_length
from modules.make_tree import make_tree

```

```

def test_1():
    parents = [-1]
    leafs = make_tree(parents)
    assert find_length(leafs, parents) == 1

```

```

def test_2():
    parents = [-1, 0, 4, 0, 3]
    leafs = make_tree(parents)
    assert find_length(leafs, parents) == 4

```

```

def test_3():
    parents = [-1, 0, 0, 0, 0, 0, 0, 0]
    leafs = make_tree(parents)
    assert find_length(leafs, parents) == 2

```

```

def test_4():
    parents = [7, 0, 6, 2, 5, 1, 4, -1]
    leafs = make_tree(parents)
    assert find_length(leafs, parents) == 8

```

```
def test_5():  
    parents = [-1, 0, 1, 2, 2, 4]  
    leafs = make_tree(parents)  
    assert find_length(leafs, parents) == 5
```