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. 1304 _____ Байков Е.С.

Преподаватель _____ Глазунов С.А.

Санкт-Петербург

2022

Цель работы

Изучить основные методы вычисления высоты дерева с помощью языка программирования Python. Освоить проверку корректности кода с помощью pytest.

Задание

На вход программе подается корневое дерево с вершинами $\{0, \ldots, n-1\}$, заданное как последовательность $parent_0, \ldots, parent_{n-1}$, 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 $parent_0, \ldots, parent_{n-1}$. Для каждого $0 \le i \le n - 1$, $parent_i$ — родитель вершины i; если 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

Для поиска высоты дерева был использован принцип работы очереди. Была реализована функция *find_heigh*, которая принимала построенное дерево *tree* и корень дерева *root*. Внутри функции был создан список *queue*, который имитирует очередь, была объявлена переменная хранящая значение высоты дерева *length* равная нулю. Цикл *while* будет работать до тех пор, пока в переменной *root* не окажется пустой список, что означает, что программа дошла до самого дальнего листа дерева. В цикле мы присваиваем списку *queue* значение *root*, а *root* будет ссылаться на пустой список. Цикл *for* проходит по всем элементам списка *queue*, и за каждую итерацию цикла в *root* будет добавляться из дерева значение детей данного элемента и увеличиваться значение высоты *length* на единицу. Функция возвращает *length*.

Реализована функция *make_children_list*, принимающая на вход *parents* в виде строки или списка, а также *n* количество узлов. Внутри функции создается список *children_list*, который хранит в *n+1* список. С помощью условного оператора *if* идет проверка на тип переменной *parents*. В случае, если переменная является строкой также идет проверка на пустую строку и функция возвращает список пустых списков. Иначе преобразует строку в список с помощью деления строки по пробелам функцией *split*. Затем список *children_list* заполняется с помощью цикла *for*. В значение с индексом *parent_new[i]* или *parant[i]* будут занесены значения *i*. Затем функция вернет список *children_list*.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования.

№ п/п	Входные данные	Выходные данные	Комментарии
1.	10 9 7 5 5 2 9 9 9 2 -1	4	-
2.	5 4 -1 4 1 1	3	-
3.	5 -1 0 4 0 3	4	-
4.	1 -1	1	-
5.	0	0	-

Выводы

Изучены основные методы взаимодействия с древом. Изучено несколько способов нахождения высоты заданного древа. Реализована программа по нахождению высоты дерева с помощью структуры данных - очередь. Проведена проверка кода с помощью pytest.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
def find_heigh(tree, root):
    queue = []
    length = 0
    while root:
        queue = root
        root = []
        for elem in queue:
            root += tree[elem]
        length += 1
    return length

def make_children_list(parents, n):
    children_list = [[] for x in range(n+1)]
    if type(parents) == list:
        for i in range(n):
            children_list[parents[i]].append(i)
    elif type(parents) == str:
        if parents == '':
            return children_list
        parents_list = list(map(int, parents.split(' ')))
        for i in range(n):
            children_list[parents_list[i]].append(i)
    return children_list

if __name__ == '__main__':
    n = int(input())
    parents = input()

    print(find_heigh(make_children_list(parents, n),
make_children_list(parents, n)[-1]))
```

Название файла: tests.py

```
from main import find_heigh, make_children_list

def test():
    children = make_children_list([9, 7, 5, 5, 2, 9, 9, 9, 2, -1],
10)
    assert find_heigh(children, children[-1]) == 4
    children = make_children_list([4, -1, 4, 1, 1], 5)
    assert find_heigh(children, children[-1]) == 3
    children = make_children_list([-1, 0, 4, 0, 3], 5)
    assert find_heigh(children, children[-1]) == 4
    children = make_children_list('-1', 1)
    assert find_heigh(children, children[-1]) == 1
    children = make_children_list('', 0)
```

```
assert find_heigh(children, children[-1]) == 0
```