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

ОТЧЕТ

по лабораторной работе № 1

по дисциплине «Алгоритмы и структуры данных»

Тема: Вычисление высоты дерева

Студент гр. 0383 _____ Тарасов К.О.

Преподаватель _____ Берленко Т.А.

Санкт-Петербург 2021

Цель работы

Изучить механизм работы дерева, реализовать алгоритм вычисления высоты дерева, научиться тестировать программу

Задание

Вычисление высоты дерева. Python

На вход программе подается корневое дерево с вершинами {0, . . . , n−1}, заданное как последовательность parent0 , . . . , parentn−1 , где parenti —

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 parent0 , ... , parentn−1. Для каждого $0 \leq i \leq n - 1$, parenti — 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.

Ход работы

Сначала мы должны прочитать входные данные: длина массива родителей и массив родителей. Далее проходим по массиву родителей и ищем высоту графа на каждом узле и записываем его в массив depth, после находим максимальное значение в этом массиве — оно и будет высотой дерева.

Описание всех функций, использующихся в программе:

1. input_data() - функция для ввода всех данных

2. findHeight(parent, n) – на выход принимает родительский и массив и его длину. Функция проходит по массиву родителей parent и передаёт очередное значение в функцию findDepth, после находит максимальное значение в массиве высот depth и возвращает его

3. findDepth(parent, i, depth) – на вход принимает массив родителей, номер элемента и массив высот узлов, вычисляет высоту узла и изменяет массив depth в соответствии с найденной велечиной

4. findHeightTest() - функция для тестирования findHeight

Тестирование

Результаты тестирования представлены в таблице:

Табл. 1 — Результаты тестирования

Номер	Входные данные	Выходные данные	Комментарии
1	5 4 -1 4 1 1	3	Верно
2	5 -1 0 4 0 3	4	Верно
3	2 -1 0	2	Верно
4	5 -1 0 0 0 2	3	Верно
5	8 -1 0 1 2 3 4 5 6 7	8	Верно
6	6 3 3 3 -1 3 3	2	Верно

Выводы.

Был изучен механизм работы дерева, реализован алгоритм вычисления высоты дерева, программа была протестирована

ПРИЛОЖЕНИЕ А

Текст компонентов программы hello1.exe

src/main.py:

```
from tests.findHeightTest import findHeightTest
from src.input import input_data
from src.findHeight import findHeight

if __name__ == '__main__':
    findHeightTest()
    n, parent = input_data()
    print(findHeight(parent, n))
```

src/input.py:

```
# Ввод данных
def input_data():
    n = int(input())
    parent = list(map(int, input().split()))
    return n, parent
```

src/findHeight.py:

```
from src.findDepth import findDepth



# Нахождение высоты дерева
def findHeight(parent, n):
    depth = [0] * n

    for i in range(n):
        findDepth(parent, i, depth)
```

```
    m = depth[0]
    for i in range(1, n):
        if depth[i] > m:
            m = depth[i]

    return m
```

src/findDepth.py:

```
# Нахождение высоты узла дерева
def findDepth(parent, i, depth):
    if parent[i] == -1:
        depth[i] = 1
        return

    if depth[i] != 0:
        return

    if depth[parent[i]] == 0:
        findDepth(parent, parent[i], depth)

    if depth[parent[i]] != 0:
        depth[i] = depth[parent[i]] + 1
        return
```

tests/findHeightTest.py:

```
from src.findHeight import findHeight


def findHeightTest():
    assert (findHeight([-1, 0], 2) == 2)
    assert (findHeight([4, -1, 4, 1, 1], 5) == 3)
    assert (findHeight([-1, 0, 4, 0, 3], 5) == 4)
```

```
assert (findHeight([-1, 0, 0, 0, 2], 5) == 3)
assert (findHeight([-1, 0, 1, 2, 3, 4, 5, 6, 7], 8) == 8)
assert (findHeight([3, 3, 3, -1, 3, 3], 6) == 2)
```