**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. 1384 |  | Усачева Д.В. |
| Преподаватель |  | Берленко Т.А. |
|  |  |  |

Санкт-Петербург 2022

**Цель работы.**

Реализация алгоритма обхода графа для нахождения высоты дерева.

**Задание.**

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 ≤ i ≤ 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.*

**Выполнение работы.**

Была разработана функция height\_t(n, tree)),принимающая ссылку на дерево и количество элементов. Возвращает эта функция количество вершин в самом длинном пути от корня к листу.

Разработанный программный код см. в приложении 2.

Результаты тестирования см. в приложении 1.

**Выводы.**

Был использован алгоритм обхода графа для вычисления высоты переданного дерева.

**ПРИЛОЖЕНИЕ 1**

**ТЕСТИРОВАНИЕ**

Таблица 1 - Пример тестовых случаев.

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
| 1. | 1  -1 | 1 |
| 2 | 7  -1 0 0 0 3 4 2 | 4 |
| 3 | 5  4 -1 4 1 1 | 3 |
| 4 | 12  -1 0 1 2 3 4 4 4 4 6 6 6 | 7 |
| 5 | 3  -1 0 2 | 2 |

**ПРИЛОЖЕНИЕ 2**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: main.py

#python

def height\_t(n, tree):

arr\_hg = [0] \* n

for i in range(n):

parent = i

while parent != -1:

parent = tree[parent]

arr\_hg[i] += 1

if arr\_hg[parent]:

arr\_hg[i] += arr\_hg[parent]

break

mx\_hg=max(arr\_hg)

return mx\_hg

n = int(input())

str\_tree = input()

tree = list(map(int, str\_tree.split()))

height = height\_t(n, tree)

print(height)

Название файла: tests.py

from main import height\_t

def tests():

assert height(5, [4, -1, 4, 1, 1]) == 3

assert height(7, [1, 0, 0, 0, 3, 4, 2]) == 4

assert height(12, [-1, 0, 1, 2, 3, 4, 4, 4, 4, 6, 6, 6]) == 7

assert height(3, [-1, 0, 2]) == 2

assert height(1, [-1]) == 1

if \_\_name\_\_ == '\_\_main\_\_':

tests()

print('Everything is fine!')