

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

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Деревья**

Студент гр. 9305

\_\_\_\_\_

Николаенко К. Н.

Преподаватель

\_\_\_\_\_

Манирагена В.

Санкт-Петербург

2020

### Цель работы.

Исследование алгоритмов для работы с двоичным деревом.

### Задание.

Нужно вычислить высоту левого поддерева относительно корня.

### Обоснование выбора представления дерева.

Исходя из 1 и 2 лабораторной работы, можно сделать вывод, что удобнее всего работать со списками, так как в них можно осуществить достаточно большой функционал, Так же не маловажно что списки используют указатели на свои соседние элементы, что при работе с памятью более удобно, чем массив, ну и списки могут ветвиться.

### Тестовый пример.

```
0)Random
1)Manually
Input mode: 1
Node(a,0) 1/0:1
Node(a,1) 1/0:1
Node(a,2) 1/0:0
Node(a,2) 1/0:1
Node(a,3) 1/0:1
Node(a,4) 1/0:1
Node(a,5) 1/0:0
Node(a,5) 1/0:0
Node(b,4) 1/0:0
Node(c,3) 1/0:0
Node(e,1) 1/0:1
Node(e,2) 1/0:0 .....h.....
Node(e,2) 1/0:1 .d.....g.
Node(e,3) 1/0:1 .....c.....f.
Node(e,4) 1/0:0 .....b.....e.
Node(e,4) 1/0:0 ....a.....
Node(f,3) 1/0:0 .....
```

## Контрольные тесты.

0)Random

1)Manually

Input mode: 0

```
.....g.....
.....f.....
.....e.....
...a.....d.....
.....c.....
.....b.....
.....
.....
```

Depth-first search:g f e a d c b Passed nodes: 7

Height of the left subtree = 4

0)Random

1)Manually

Input mode: 0

```
.....h.....
.....f.....g.....
.....e.....
.....d.....
.....b..c.....
.....a.....
.....
.....
```

Depth-first search:h f e d b a c g Passed nodes: 8

Height of the left subtree = 4

**Результаты измерения времени обработки каждым из способов.**

№	функция	временная сложность
1	создание дерева	$O(V)$
2	обход в глубину	$O(V)$
3	вывод	$O(V)$

Где  $V$  - количество вершин (множество вершин)

### **Выводы о результатах испытания алгоритмов обхода дерева.**

Проделав данную работу, я пришел к выводу, что алгоритм в глубину является достаточно хорошим для обхода дерева, обход в ширину тоже не плох, но мне почему то ближе алгоритм обхода в глубину. Мне понравилась реализация стека и очереди. Так же хочу отметить, что оба алгоритма достаточно простые для понимания.

В общем, хочу сказать, что работа была достаточно интересная, особенно процесс создания дерева, однако вывод дерева был не сразу понятен мне, но я все же разобрался.

### **Список использованных источников.**

1. Пользовательские структуры данных. Методические указания по дисциплине «Алгоритмы и структуры данных, часть 1» П. Г. Колинко, 2020.

### **Приложение А.**

Ссылка на гитхаб с кодом:

[https://github.com/NikolaenkoKonstantin/AiSD/tree/main/lab\\_3](https://github.com/NikolaenkoKonstantin/AiSD/tree/main/lab_3)