Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**Отчёт по заданию № 3**

**на тему: “Деревья”**

**по дисциплине**

**“Алгоритмы и структуры данных”**

**Вариант 26**

|  |  |
| --- | --- |
| Выполнили студент гр.9308: | Яловега Н.В. |
| Проверил: | Колинько П.Г. |

Санкт-Петербург, 2020 г.

Оглавление

[Введение 3](#_Toc54989561)

[1. Задание 3](#_Toc54989562)

[2. Формализация задания 3](#_Toc54989563)

[3. Пример вывода дерева 4](#_Toc54989564)

[4. Временная сложность функций обхода. 5](#_Toc54989565)

[5. Результаты прогона программы. 5](#_Toc54989566)

[Вывод 7](#_Toc54989567)

[Список используемых источников 8](#_Toc54989568)

[Приложение 1 (Исходный текст программы) 9](#_Toc54989569)

## Введение

Исследование алгоритмов для работы с троичным деревом.

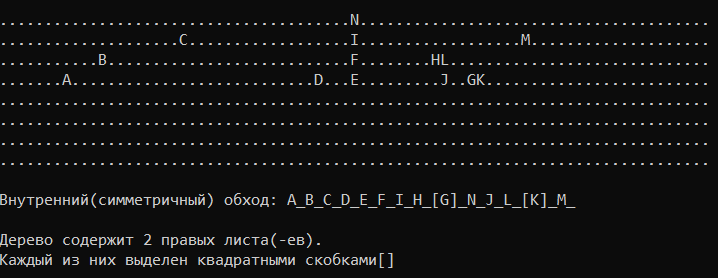
## 1. Задание

Требуется создать троичное дерево с симметричной разметкой узлов. Используя способ обхода в глубину, подсчитать количество листьев не на самом нижнем уровне исходного дерева.

## 2. Формализация задания

Было принято представлять дерево в памяти ЭВМ с помощью разветвляющегося списка. ...

## 3. Пример вывода дерева

Пример вывода дерева при выполнении программы:  
  


*Рисунок 1. Пример вывода дерева.*

Ввод обозначения узла символом осуществляется обратным образом. Это можно наглядно увидеть, если посмотреть, что самый левый лист обозначается буквой «A» (обозначение узлов начинается с буквы «A»). Чтобы при выводе троичного дерева не создавалось перекрытие элементов, они выводятся на отдалении в один символ от центра(например, H - правый сын I, а L - левый сын M). Так же ниже представлен обратный внутренний обход дерева. Сначала обходится левое поддерево, затем среднее поддерево, после обрабатывается корень, и в последнюю очередь обрабатывается правое поддерево.

## 4. Временная сложность функций обхода.

1. **Создание дерева.**

Сложность: Создание дерева - добавление в изначально пустое дерево *n* элементов имеет временную сложность O(*n-1*).

1. **Обход дерева**Сложность: О(*n*), так как это рекурсивный обход по n узлам дерева
2. **Обработка дерева**

Сложность: О(*n*), так как правые листья подсчитываются непосредственно в момент обхода дерева по n узлам.

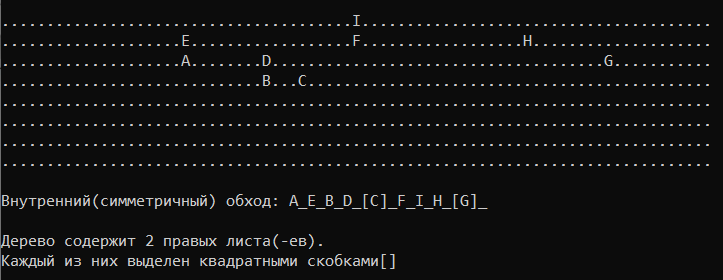
1. **Вывод дерева**

Сложность: О(*n*), так как мы просто обходим дерево.

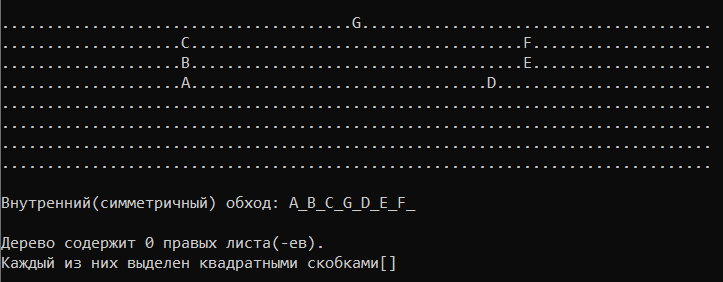
## 5. Результаты прогона программы.

При работе программы выводится количество правых листьев у исходного дерева.

Также выводится порядок обработки элементов дерева при внутреннем(симметричном) обходе.

На скриншотах ниже приведены различные ситуации прогона программы.  
  


*Рис 2. Результат выполнения.*

  
*Рис 3. В дереве отсутствуют правые листья.*

## Вывод

* + - * 1. Структура «Дерево» одна из базовых и необходимых инструментов в программировании. Это довольно простой вариант представления каких-либо связных между собой данных, поиска в этой базе, а что куда важнее для программирования - высокая скорость работы с этими базами. «Дерево» основанное на узлах и указателях мало того, что интуитивно понятно(узел - элемент дерева, указатель - связующая линия двух элементов), так еще и имеет большое преимущество по скорости работы перед «Деревом», основанном на массиве или векторе.

Выполнение данной лабораторной поможет в дальнейшей реализации структуры данных «Граф», так как это всеобъемлющее понятие, включающее в себя в том числе и «Дерево».

## Список используемых источников

1. Колинько П.Г. Пользовательские структуры данных / Методические указания по дисциплине «Алгоритмы и структуры данных» - Санкт-Петербург: СПбГЭТУ «ЛЭТИ», 2020.

## Приложение 1 (Исходный текст программы)