|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

**«ОБРАБОТКА ДЕРЕВЬЕВ И ХЕШ-ФУНКЦИЙ»**

Студент Лазутин Александр Владимирович

Группа ИУ7 – 33Б

*2022*

Оглавление

[Описание условия задачи 3](#_Toc121153796)

[Описание технического задания 3](#_Toc121153797)

[Описание алгоритма 5](#_Toc121153798)

[Набор тестов 5](#_Toc121153799)

[Оценка эффективности 7](#_Toc121153800)

[Ответы на контрольные вопросы 8](#_Toc121153801)

[Вывод 9](#_Toc121153802)

# **Описание технического задания**

В текстовом файле содержатся целые числа. Построить двоичное дерево из чисел файла. Вывести его на экран в виде дерева. Используя процедуру, определить количество узлов дерева на каждом уровне. Добавить число в дерево и в файл. Сравнить время добавления чисел в указанные структуры.

Входные данные:

1. Целое число, представляющее собой пункт меню:

целое число в диапазоне от 0 до 3

2. Дополнительный ввод: поле типа int в зависимости от команды

Выходные данные:

1. Результат выполнения команды

2. Сообщение об ошибке (при ее возникновении)

Функции программы:

1. Ввести данные из файла

2. Вывести двоичное дерево поиска

3. Добавить элемент в двоичное дерево и файл

0. Выйти из программы

Аварийные ситуации:

1. Неверно введен пункт меню

(не число или число меньшее 0 или больше 3)

2. Введенный файл не существует

(неверное имя файла)

3. Неверно введено число в файле

(не число)

4. Число, которое нужно добавить уже содержится в структурах программы

Описание структуры данных

Структура для хранения массива чисел, считанных из файла

typedef struct arr\_r

{

int \*data;

int len;

int capacity;

int max\_cap;

} arr\_t;

Поля структуры:

1. int \*data— массив значений из файла

2. int len — длина массива чисел

3. int capacity — текущий объем памяти, задействованной в массиве

4. int max\_cap - максимальный объем памяти, выделенной под массив

typedef struct node\_r

{

int data;

unsinged char height;

struct node\_r \*left;

struct node\_r \*right;

} node\_t;

Поля структуры:

1. int data — значение текущего корня

2. unsinged char height — высота вершины относительно других вершин

3. struct node\_r \*left— указатель на левого предка корня

4. struct node\_r \*right— указатель на правого предка корня

# 

# **Описание алгоритма**

1. Выводится меню программы.

2. Пользователь вводит номер любой команды, которой соответствует свое назначение.

3. Ввод осуществляется, пока не будет совершена ошибка при вводе или не будет введен 0.

# **Набор тестов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название теста** | **Пользовательский ввод** | **Результат** |
| 1 | Некорректный ввод пункта меню | aaa | Ошибка: пункты меню это числа от 0 до 3 |
| 2 | Нет файла, соответствующего введенному имени | имя несуществующего файла | Ошибка: неверно введено имя файла |
| 3 | Число в файле введено неверно (не число) | aaa | Ошибка: неверно число в файле |
| 4 | Невозможно добавить два одинаковых числа в структуры данных | Попытка добавить уже имеющееся число в структурах данных | Ошибка: такое число уже существует |
| 5 | Неверное число при добавлении | aaa | Ошибка: неверно введено число для добавления |
| 6 | Ввод данных из файла | Команда 1  корректное имя файла | Данные из файла введены успешно |
| 7 | Вывести дерево двоичного поиска | Команда 2  данные из файла введены | Вывод двоичного дерева |
| 8 | Добавление элемента | Команда 3 | Вывод замеров времени, памяти, сравнений для добавления элемента в файла |
| 9 | Выход из программы | Команда 0 | Выход из программы, очистка консоли |

# **Оценка эффективности**

Добавление элементов (в тиках)

|  |  |  |
| --- | --- | --- |
| Количество элементов | Двоичного дерево | Файл |
| 10 | 1242 | 26352 |
| 100 | 1762 | 29195 |
| 500 | 2278 | 31978 |
| 1000 | 2567 | 33777 |

Приведены средние значения

Память

|  |  |  |
| --- | --- | --- |
| Количество элементов | Двоичного дерево | Файл |
| 10 | 264 | 31 |
| 100 | 2400 | 289 |
| 500 | 12024 | 1882 |
| 1000 | 24024 | 3763 |

# 

# **Ответы на контрольные вопросы**

*1. Что такое дерево?*

Дерево — структура данных (рекурсивная), используемая для представления иерархических связей (один ко многим)

*2. Как выделяется память под представление деревьев?*

Память выделяется как для связанного списка, то есть под каждый узел отдельно.

*3. Какие бывают типы деревьев?*

ДДП и АВЛ

*4. Какие стандартные операции возможны над деревьями?*

Поиск по дереву, обход дерева, добавление элемента в дерево, удаление элемента из дерева

*5. Что такое дерево двоичного поиска?*

Двоичное дерево поиска (ДДП) — двоичное дерево

В нем для каждого узла выполняется условие, что левый потом больше или равен родителю, а правый потомок строго меньше родителя (или наоборот)

# **Вывод**

В данной лабораторной работе сравнивается добавление элемента в 2 разных структуры данных

В двоичного дерево элементы добавляются гораздо быстрее, чем в файл, но оно проигрывает по памяти в ~8 раз.