**МГТУ им. Н.Э.Баумана**

**Дисциплина “Типы и структуры данных”**

**Отчет по Лабораторной работе №6 по теме “** **Деревья,**

**хеш –таблицы** **”.**

Работу выполнил:

Студент: Зейналов Зейнал

ИУ7-31Б

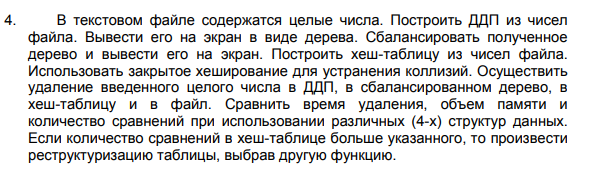
Варинат 14(4)

Москва, 2018.

**Работа № 6**

**Цель работы – построить дерево, вывести его на экран в виде дерева, реализовать основные операции работы с деревом: обход дерева, включение, исключение и поиск узлов, сбалансировать дерево, сравнить эффективность алгоритмов сортировки и поиска в зависимости от высоты деревьев и степени их ветвления; построить хеш-таблицу и вывести ее на экран, устранить коллизии, если они достигли указанного предела, выбрав другую хеш-функцию и реструктуризировав таблицу; сравнить эффективность поиска в сбалансированных деревьях, в двоичных деревьях поиска (ДДП) и в хеш-таблицах. Сравнить эффективность реструктуризации таблицы для устранения коллизий с эффективностью поиска в исходной таблице.**

Описание условия задачи



Описание ТЗ

Общая концепция системы

**Программа выполняет операции над ДДП в 4 структурах (Файл, сбалансированное дерево, не сбалансированное дерево, хеш-таблица).**

**Пользователь может удалить число из структур, введя само это число и минимальное количество сравнений. В случае необходимости, будет произведена реструктуризация хеш-таблицы(изменение хеш-функции).**

### Требования к функциональным характеристикам

При разработке интерфейса программы следует предусмотреть:

• указание типа, формата и диапазона вводимых данных,

• указание действий, производимых программой,

• наличие пояснений при выводе результата,

• вывод дерева осуществить в графическом виде (или предложить иную визуализацию в виде дерева)

• вывод на экран хеш-таблицы;

• возможность изменения хеш-функции при необходимости реструктуризации таблицы;

• вывод времени и количества сравнений при поиске одних и тех же данных в различных структурах данных.

При тестировании программы необходимо:

o проверить правильность ввода и вывода данных (т.е. их соответствие требуемому типу и формату). Обеспечить адекватную реакцию программы на неверный ввод данных;

o обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»);

o проверить правильность выполнения операций; o обеспечить возможность добавления узла в пустое дерево;

o предусмотреть вывод сообщения при попытке удаления узла из пустого дерева;

o проверить различные варианты включения и исключения узла в существующее дерево;

o проверить поиск существующего узла и поиск несуществующего узла в дереве и хеш-таблице;

o создать левостороннее или правостороннее дерево, проверить время обработки узла в нем;

o предусмотреть вывод сообщения при необходимости реструктуризации хеш-таблицы.

На **вход** программа получает название текстового файла, в котором хранится граф.

Затем число, которое необходимо удалить и минимально допустимое количество сравнений.

**Выход** должен быть представлен в виде набора данных, представляющих собой 4 структуры.

Также программа должна вывести данные о времени выполнения алгоритмов.

Программа должна выдавать корректный ответ при вводе любых данных. Если произошла ошибка ввода, программа должна сообщить об этом.

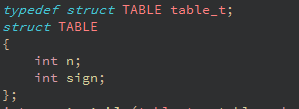
## Способ обращения к программе

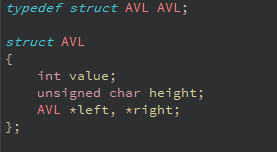
Программа представляет собой файл app.exe. Запускается в консоли. Для запуска достаточно команды ./app.exe. Если файл отсутствует можно собрать его с помощью утилиты make app.exe .

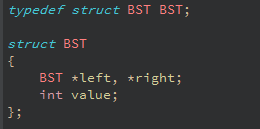
Аварийные ситуации:

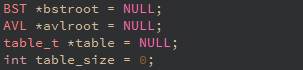
* Ввод несуществующего в данной директории файла  
  Программа выведет информацию о неверном вводе названия файла
* Введение неверного минимального количества сравнений   
  Программа выведет информацию об ошибке ввода

Описание структур данных











Описание алгоритма

Входные данные

Сравнение результатов

Оценка эффективности

Выводы по проделанной работе

# Ответы на вопросы

1. Что такое дерево?

2. Как выделяется память под представление деревьев?

3. Какие стандартные операции возможны над деревьями?

4. Что такое дерево двоичного поиска?

5. Чем отличается идеально сбалансированное дерево от АВЛ дерева?

6. Чем отличается поиск в АВЛ-дереве от поиска в дереве двоичного поиска?

7. Что такое хеш-таблица, каков принцип ее построения?

8. Что такое коллизии? Каковы методы их устранения.

9. В каком случае поиск в хеш-таблицах становится неэффективен?

10. Эффективность поиска в АВЛ деревьях, в дереве двоичного поиска и в хештаблицах