Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра МКИТ

Отчет по лабораторной № 2

по дисциплине

«Структуры и алгоритмы обработки данных»

Выполнил: студент группы

БВТ1904

Сушков И.А.

Руководитель:

Павликов А.Е.

Москва
2021

Цель работы:

Реализовать методы поиска в соответствии с заданием. Организовать генерацию начального набора случайных данных. Для всех вариантов добавить реализацию добавления, поиска и удаления элементов. Оценить время работы каждого алгоритма поиска и сравнить его со временем работы стандартной функции поиска, используемой в выбранном языке программирования.

Код (кратко):

В классе laba2Runner происходит запуск программы
В классе BinaryTree существует метод поиска B_search, удаления – метод remove, вставки – метод add.
В классе HashTable методы hash(), simple_rehash() и fake_rehash(), реализующие хеширование, простое рехеширование и рехеширование псевдослучайными числами, а так же методы add, remove для вставки и удаления, соответственно.

Результат выполнения программы:

Для массива и бинарного дерева на 30000 элементов, заполненного рандомными числами, среднее время поиска в мс:

```
Время поиска по бинарному дереву: 0
Время бинарного поиска: 10
Время интерполяционного поиска: 10
Время фиббоначиева поиска : 20
```

Время выполнения задания 2

```
Задание №2:

Вставка

Время вставки при простом хэшировании: 21
Время вставки при псевдо рехэшировании: 40
Время вставки в методе цепочек: 40
Время вставки в двоичном дереве 220

Поиск

Время поиска при простое хэширование: 10
Время поиска при псевдо рехэширование: 30
Время поиска при методе цепочек: 10

Удаление

Время удаления при простом хэшировании: 10
Время удаления при псевдо рехэшировании: 30
Время удаления при методе цепочек: 10
Время удаления из бинарного дерева: 40
```

Вывод:

В ходе выполнения данной лабораторной работы я убедился на примерах, что среди на небольших данных все поиски показывают приблизительно одинаковый результат, но с увеличением объема данных появляются различия. Самыми быстрыми оказались поиск по бинарному дереву и интерполяционный. При хешировании – самое быстрое простое.