Федеральное государственное бюджетное образовательное учреждение высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |  |
| --- | --- | --- |
| ФАКУЛЬТЕТ | ИУ | ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ |
| КАФЕДРА | ИУ6 | КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ |

**Отчет**

**по лабораторной работе №1**

**Дисциплина: Технологии разработки программных систем**

**Название: Исследование структур и методов обработки данных**

**Вариант 12**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Студент гр. ИУ6-42** |  | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **М.А. Мотичев** |
|  |  | (Подпись, дата) | | (И.О. Фамилия) |
| **Преподаватель** |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | **К.С. Хорунжина** |
|  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва 2018

# 1. ВВЕДЕНИЕ

## 1.1. Цель работы

* Исследование структур данных, методов их обработки и оценки.

## 1.2. Задача

### 2.1. Порядок выполнения

1. Ознакомиться с теоретическими сведениями по абстрактным структурам данных и методами их обработки.
2. Для указанной задачи и типа данных (см. таблицу 2) предложить способ реализации и определить требуемый объем памяти.
3. Провести анализ заданных методов поиска, упорядочения и корректировки. Оценить время выполнения соответствующих операций.
4. Предложить альтернативный вариант решения задачи, в котором должно быть минимум одно улучшение. Улучшения могут касаться как структуры данных, так и основных операций. Обосновать новые решения, используя количественные и качественные критерии. Количественными критериями являются: объем памяти, среднее количество сравнений и количество тактов. Качественные критерии определяют возможность использования того или иного метода применительно к разработанной структуре. К ним можно отнести: применимость операции только к упорядоченным данным; необходимость знать количество элементов; наличие признака разбивки на гнезда; необходимость в прямом доступе к элементам; знание граничных значений; невозможность создать структуру в соответствии с арифметической прогрессией и др.

## 1.3. Задание

Для варианта 12 задание выглядит следующим образом:

Задача 3. Даны M записей вида: код группы; ФИО; дата рождения;

* Структура: Таблица;
* Поиск: Дихотомический (двоичный);
* Упорядочение: Вставкой;
* Корректировка: Удаление маркировкой.

# 2. ОСНАВНАЯ ЧАСТЬ

## 2.1. Таблица исходных данных

Таблица 1. Исходные данные.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | Задача | Структура | Поиск | Упорядочение | Корректировка |
| 12 | 3 | Таблица | Дихотомич. | Вставкой | Удаление маркир. |

## 2.2. Постановка задачи

Задача 3. Даны M записей вида: код группы; ФИО; дата рождения;

Исходя из исходных данных и поставленной задачи следует то, что необходимо:

* Составить структуру для записи;
* Заполнить таблицу, которая будет из себя представлять одномерный массив для реализации заданных методов;
* Реализовать заданные методы;
* Провести анализ. Для этого взять несколько значений М для более точного анализа.

## 2.3. Анализ исходных данных

### 2.3.1. Определение объёма данных

Для удаления маркировкой необходимо добавить в структуру поле, отвечающее за маркер. Полученная структура данных представлена на рисунке 1.

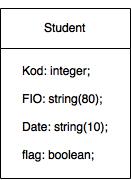


Рис.1. Структура данных.

Объём данных для задачи задан как М => возьмём несколько значений М для более точной оценки методов.

Объём структуры: Kod = 2, FIO = 80\*1, Date = 10\*1, flag = 1; Сумма = 93.

Общий объём структуры данных = М\*93.

### 2.3.2. Анализ алгоритма поиска

По заданию необходимо использовать дихотомический способ поиска. Для его эффективного применения необходимо, чтобы массив был заранее отсортирован.

а) Массив заранее отсортирован

В этом случае просто применяем код, представленный в Листинге 1, чтобы отсортировать наш массив.

Листинг 1. Поиск дихотомическим способом.

код

В итоге получаем следующие результаты:

1. При М =
2. При М =
3. При М =

б) Массив нуждается в сортировке

При этом сложность поиска увеличивается на столько – сколько займёт алгоритм упорядочения.

### 2.3.3. Анализ алгоритма упорядочения

По заданию необходимо использовать сортировку (упорядочение) методом вставки. Метод вставки один из самых быстрых способов сортировки массива. Фрагмент кода с сортировкой методом вставки представлен в Листинге 2.

Листинг 2. Упорядочение методом вставки.

код

Вычислительная сложность будет равна:

1. При М =
2. При М =
3. При М =

### 2.3.4. Анализ алгоритма удаления.

По заданию необходимо использовать метод удаление маркировкой. Метод заключается не в удалении, а в отмечании удалённых элементов. Это не очень удобно по памяти и по вычислительной сложности для дальнейшей работы, так как при работе с элементом необходима дополнительная проверка.

## 2.4. Вывод

# 3. АЛЬТЕРНАТИВНЫЙ ВАРИАНТ

## 3.1. Таблица исходных данных

## 3.2. Анализ исходных данных

### 3.2.1. Определение объёма данных

### 3.2.2. Анализ алгоритма поиска

### 3.2.3. Анализ алгоритма упорядочения

### 3.4.2.4. Анализ алгоритма удаления.

## 3.3. Вывод

# 4. СРАВНИТЕЛЬНАЯ ТАБЛИЦА

# 5. ВЫВОД