

## **Лабораторная работа №1**

### **Абстрактные структуры данных**

**Цели и задачи работы:** изучение алгоритмов формирования и работы с абстрактными структурами данных.

**Задание к работе:** Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

**Языки программирования:** C, C++, Rust, Go

**Методика выполнения работы:**

1. Разработать алгоритм решения задачи по индивидуальному заданию.
2. Написать и отладить программу решения задачи (C, Go или Rust).
3. Протестировать работу программы на различных исходных данных.
4. По запросу преподавателя быть готовым модифицировать алгоритм и добавить операцию работы с данными.

#### **Перечень понятий к защите лабораторной работы 1.**

Понятие алгоритма. Свойства алгоритма. Алгоритм Евклида. Математический анализ алгоритма. Временная сложность алгоритма. Асимптотическая нотация. Оценки скорости роста функции. Базовые приемы при анализе программы. Основные классы алгоритмов по временной сложности. Графическое представление скорости роста алгоритмов. Исполнители алгоритмов.

Обоснование выбора алгоритма. Тип данных. Скалярные типы данных. Структурированные типы данных. Фундаментальные структуры. Динамические структуры. Концепция абстрактного типа данных. Понятие структуры данных. Инкапсуляция. Интерфейс. Внутренняя реализация. Интерфейс абстракции. Классификация структур хранения данных. Понятие линейного списка. Последовательность формирования АД в виде линейного списка. Классификация линейных списков (достоинства и недостатки разных видов линейных списков).

Соглашения при организации связанного списка. Эффективность связанного списка (доступ, поиск, вставка, удаление).

Стек. Эффективность стека на основе связанного списка (добавление, удаление, поиск).

Очередь. Эффективность очереди (добавление, удаление).

Хеш-таблица. Эффективность хеш-таблицы. Виды коллизий и алгоритмы по их устранению

Бинарное дерево. Бинарное дерево поиска. Виды обхода деревьев. Full Binary Tree. Complete Binary Tree. Сбалансированное двоичное дерево (АВЛ-дерево). Эффективность деревьев. В-дерево.

*Ссылки на сайты с визуализациями:*

<https://clck.ru/35opVK>

<https://clck.ru/35opWV>, <https://clck.ru/LS6WD>

## Реализация абстрактных структур данных.

1. Реализовать структуры данных с базовым набором операций:
  - a. Массив. Операции: добавление элемента (по индексу и в конец массива), получение элемента по индексу, удаление элемента по индексу, замена элемента по индексу, длина массива, чтение.
  - b. Список (односвязный, двусвязный). Операции: добавление элемента в голову и в хвост, удаление элемента с головы и с хвоста, удаление элемента по значению, поиск элемента по значению, чтение.
  - c. Очередь. Операции: добавление и удаление элемента (push и pop), чтение.
  - d. Стек. Операции: добавление и удаление элемента (push и pop), чтение.
  - e. Хеш таблицы. Операции: предусмотреть работу с коллизиями добавление элемента (ключ-значение), получение значения элемента по ключу, удаление элемента по ключу.
  - f. Деревья (вариант 1<sup>1</sup> Complete Binary Tree, вариант 2 Сбалансированное двоичное дерево (АВЛ-дерево), вариант 3 Full Binary Tree). Операции: добавление элемента, поиск элемента, удаление элемента с сохранением структуры дерева, чтение. Если Full Binary Tree или Complete Binary Tree, то вместо удаления элемента добавить операцию проверки на Full или Complete соответственно.
2. Реализовать интерфейс работы со структурами данных. Предусмотреть считывание из файла и запись в файл при внесении изменений в состав элементов.

Часть операций для реализации представлено в таблице. Дополнить список операций согласно п.1 и сохраняя стиль написания команды (первая буква – принадлежность структуре данных). **Исключение: Операция PRINT выводит любую структуру данных на экран.**

*Примерная таблица операций*

Тип контейнера	Добавление	Удаление	Чтение
Массив (M)	MPUSH	MDEL	MGET
Список (L)	LPUSH	LDEL	LGET
Очередь (Q)	Q PUSH	QPOP	QPOP
Стек (S)	SPUSH	SPOP	SPOP
Хеш-таблица (H)	HSET	HDEL	HGET
Дерево (T)	TINSERT	TDEL	TGET

3. Обосновать сложность выполнения каждой операции с позиции BigO нотации. Составить таблицу BigO по операциям и структурам в отчете.
4. В отчете в контрольном примере представить визуальные формы, подтверждающие соответствие результатам работы программы.

<sup>1</sup> Номер варианта выбирается по модулю 3 по списку группы. При желании можно вместо своего варианта реализовать В-дерево.

## Пример

```
./dbms --file file.data --query 'HSET myhash key value'
```

### Разбор примера

- ./dbms имя нашей программы, используя ./ запускаем программу
- --file file.data используя ключ --file указываем файл в котором содержатся данные с которыми будет работать СУБД
- --query 'HSET myhash key value' используя ключ --query делаем запрос к базе данных, текст означает - положить в хэш-таблицу myhash ключ key со значением value

### Дополнительные примеры

#### Добавление элемента item в стек с именем mystack

```
./dbms --file file.data --query 'SPUSH mystack item'  
-> item
```

#### Чтение и удаление элемента из очереди с именем myqueue

```
./dbms --file file.data --query 'QPOP myqueue'  
-> element
```

#### Проверка содержится ли элемент value в множестве с именем myset

```
./dbms --file file.data --query 'SISMEMBER myset value'  
-> TRUE
```

## Литература

1. Т. Кормен. Алгоритмы. Построение и анализ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн . – Издательство «Вильямс», 2013.
2. Дж. Макконелл Анализ алгоритмов. Активный обучающий подход. — 3-е дополненное издание. М: Техносфера, 2009. -416с.
3. Миллер, Р. Последовательные и параллельные алгоритмы: Общий подход / Р. Миллер, Л. Боксер ; пер. с англ. — М. : БИНОМ. Лаборатория знаний, 2006. — 406 с.
4. Скиена С. Алгоритмы. Руководство по разработке. 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург. 2011. — 720 с.: ил.
5. Фундаментальные алгоритмы на С++ часть 1-4 (Роберт Седжвик) 2001 (или другие книги по алгоритмам данного автора).