

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Н.А. Соловьева

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

АЛГОРИТМЫ СОРТИРОВКИ

по курсу: АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4314

подпись, дата

Д. М. Развеев

инициалы, фамилия

Санкт-Петербург 2024

1. Цель работы:

Научиться реализовывать алгоритмы сортировки.

2. Вариант 18 (задания 4 и 13):

Задание 4: Реализуйте структуру данных «Массив», элементами которого выступают экземпляры класса Book (минимум 10 элементов), содержащие следующие поля (автор, издательство, кол-во страниц, стоимость, ISBN). Добавьте методы для сортировки перемешиванием (по возрастанию) по полю «стоимость» и сортировки выбором (по убыванию) по полю «кол-во страниц».

Задание 13: Реализуйте структуру данных «Массив», элементами которого выступают экземпляры класса Book (минимум 10 элементов), содержащие следующие поля (автор, издательство, кол-во страниц, стоимость, ISBN). Добавьте методы для сортировки перемешиванием (по возрастанию) по полю «ISBN» и быстрой сортировки (по убыванию) по полю «стоимость».

3. Ход работы

Задание 4

Программа реализации:

```
from typing import List
import time

# Класс книги с полями: автор, издательство, количество страниц, стоимость, ISBN
class Book:
    def __init__(self, author: str, publisher: str, pages: int, price: float, isbn: str) -> None:
        self.author = author
        self.publisher = publisher
        self.pages = pages
        self.price = price
        self.isbn = isbn

    def __repr__(self) -> str:
        return f"Book(author='{self.author}', publisher='{self.publisher}', pages={self.pages}, price={self.price}, ISBN='{self.isbn}')"

# Класс массива книг с методами сортировки, добавления, удаления и итерируемостью
class BookArray:
    def __init__(self) -> None:
        self.books: List[Book] = []

    def __repr__(self) -> str:
```

```

        return f"[{'', '.join(str(book) for book in self.books)}]"

# Метод для добавления книги
def append(self, book: Book) -> None:
    self.books.append(book)

# Метод для удаления книги по ISBN
def delete(self, isbn: str) -> None:
    self.books = [book for book in self.books if book.isbn != isbn]

# Метод сортировки перемешиванием (Shaker sort) по возрастанию цены
def cocktail_sort(self) -> None:
    left = 0
    right = len(self.books) - 1
    while left < right:
        for i in range(left, right):
            if self.books[i].price > self.books[i + 1].price:
                self.books[i], self.books[i + 1] = self.books[i + 1],
self.books[i]
        right -= 1
        for i in range(right, left, -1):
            if self.books[i - 1].price > self.books[i].price:
                self.books[i], self.books[i - 1] = self.books[i - 1],
self.books[i]
        left += 1

# Метод сортировки выбором (Selection sort) по убыванию количества страниц
def quick_sort(self) -> None:
    self.books.sort(key=lambda book: book.pages, reverse=True)

# Делает класс итерируемым
def __iter__(self):
    return iter(self.books)

def __str__(self) -> str:
    return '\n'.join(str(book) for book in self.books)

```

Приведём описание кода:

Этот код реализует структуру данных для работы с массивом объектов класса `Book`, которые представляют книги с определенными атрибутами. Структура поддерживает операции добавления, удаления, сортировки и итерирования. Вот краткое описание основных элементов:

Класс `Book`

`Book` представляет книгу и имеет следующие поля:

- **author** (строка): автор книги.
- **publisher** (строка): издательство книги.

- **pages** (целое число): количество страниц в книге.
- **price** (вещественное число): стоимость книги.
- **isbn** (строка): уникальный идентификатор ISBN книги.

Метод `__repr__` возвращает строковое представление объекта `Book` для удобного вывода информации.

Класс `BookArray`

`BookArray` представляет собой массив объектов `Book` и предоставляет методы для работы с этим массивом:

1. Метод **`append`** добавляет новый объект `Book` в массив.
2. Метод **`delete`** удаляет книгу из массива по заданному ISBN.
3. Метод **`cocktail_sort`** реализует сортировку перемешиванием (Shaker Sort) по возрастанию цены книг.
4. Метод **`quick_sort`** сортирует книги в порядке убывания количества страниц, используя стандартный метод Python `sort` с ключом и обратным порядком.
5. Метод **`__iter__`** делает класс итерируемым, чтобы можно было использовать его в циклах `for`.

А теперь проведем тесты и замер производительности:

```
# Функция для тестирования класса BookArray
def tests_array() -> None:
    print("\nТесты:")
    arr = BookArray()

    # Тест 1. Проверка добавления элементов
    books = [
        Book("Джек Лондон", "Эксмо", 323, 699.99, "7-1343-16"),
        Book("Макс Фрай", "Эксмо", 687, 450.0, "4-4551-18"),
        Book("Владимир Торин", "Миф", 767, 899.0, "1-145-17"),
        Book("Джон Мильтон", "АСТ", 445, 399.99, "6-4166-815"),
        Book("Уильям Индик", "Миф", 380, 500.0, "5-51-1998"),
        Book("Кэролайн О`Дохонью", "LikeBook", 365, 799.0, "2-190-10"),
        Book("Лия Арден", "Эксмо", 414, 1280.0, "8-21-3145"),
        Book("Михаил Булгаков", "Азбука", 1114, 1349.0, "9-5631-156"),
        Book("Гастон Леру", "Эксмо", 317, 320.0, "3-1342-2413"),
        Book("Бернар Вербер", "Эксмо", 412, 699.99, "8-3165-3614")
    ]
    print(f"Array до добавления в него элементов:\n{arr}")
    for book in books:
        arr.append(book)
    print(f"Array после добавления в него элементов:\n{arr}")

    # Тест 2. Проверка метода сортировки перемешиванием (по возрастанию цены)
    arr.cocktail_sort()
    print(f"\nArray после сортировки перемешиванием (по возрастанию цены):\n{arr}")
```

```

# Тест 3. Проверка метода быстрой сортировки (по убыванию количества страниц)
arr.quick_sort()
print(f"\nArray после быстрой сортировки (по убыванию количества страниц):\n{arr}")

# Тест 4. Удаление элементов по полю ISBN
arr.delete("2-190-10")
arr.delete("7-1343-16")
arr.delete("1-145-17")
arr.delete("8-3165-3614")
arr.delete("4-4551-18")
print(f"\nArray после удаления 5 элементов:\n{arr}\n")

# Тест 5. Проверка итерируемости
print("Проверка итерируемости:")
for book in arr:
    print(book)

print("Все тесты пройдены!")

# Функция для выполнения бенчмарков
def benchmarks() -> None:
    print("\nБенчмарки")
    arr = BookArray()

    # Создание 10,000 книг с уникальными значениями
    books = [Book(f'Author{i}', f'Publisher{i}', i + 100, i * 100.0, f'ISBN{i}')]
    for i in range(10000)]

    # Бенчмарк на добавление 10,000 элементов
    start = time.time()
    for book in books:
        arr.append(book)
    end = time.time()
    print(f"Время добавления 10,000 элементов: {(end - start):.5f} секунд")

    # Бенчмарк на сортировку перемешиванием
    start = time.time()
    arr.cocktail_sort()
    end = time.time()
    print(f"Время сортировки перемешиванием для 10,000 элементов: {(end - start):.5f} секунд")

    # Бенчмарк на сортировку по убыванию количества страниц
    start = time.time()
    arr.quick_sort()
    end = time.time()
    print(f"Время быстрой сортировки для 10,000 элементов: {(end - start):.5f} секунд")

# Запуск тестов и бенчмарков

```

```
tests_array()  
benchmarks()
```

Результаты тестов:

Тесты:

Array до добавления в него элементов:

Array после добавления в него элементов:

```
Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')  
Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')  
Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')  
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')  
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')  
Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')  
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')  
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')  
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')  
Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')
```

Array после сортировки перемешиванием (по возрастанию цены):

```
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')  
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')  
Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')  
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')  
Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')  
Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')  
Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')  
Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')  
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')  
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
```

Array после быстрой сортировки (по убыванию количества страниц):

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')
Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')
Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
```

Array после удаления 5 элементов:

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
```

Проверка итерируемости:

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
```

Все тесты пройдены!

Бенчмарки

Время добавления 10,000 элементов: 0.00000 секунд

Время сортировки перемешиванием для 10,000 элементов: 2.22555 секунд

Время быстрой сортировки для 10,000 элементов: 0.00000 секунд

Видно, что класс верно выполнил все свои функции, значит код работает корректно.

Задание 13

Программа реализации:

```
import time
from typing import List

class Book:
    def __init__(self, author: str, publisher: str, pages: int, price: float, isbn: str) -> None:
        self.author = author
        self.publisher = publisher
        self.pages = pages
        self.price = price
        self.isbn = isbn

    def __repr__(self) -> str:
        return f"Book(author='{self.author}', publisher='{self.publisher}', pages={self.pages}, price={self.price}, ISBN='{self.isbn}')"

class BookArray:
    def __init__(self) -> None:
        self.books: List[Book] = []

    def __repr__(self) -> str:
        return f"[{'', '.join(str(book) for book in self.books)}]"

    def append(self, book: Book) -> None:
        self.books.append(book)

    def delete(self, isbn: str) -> None:
        self.books = [book for book in self.books if book.isbn != isbn]

    def cocktail_sort_isbn(self) -> None:
        left = 0
        right = len(self.books) - 1
        while left < right:
            for i in range(left, right):
                if self.books[i].isbn > self.books[i + 1].isbn:
                    self.books[i], self.books[i + 1] = self.books[i + 1], self.books[i]
            right -= 1
            for i in range(right, left, -1):
                if self.books[i - 1].isbn > self.books[i].isbn:
                    self.books[i], self.books[i - 1] = self.books[i - 1], self.books[i]
            left += 1
```



```

def quick_sort_price(self) -> None:
    self.books.sort(key=lambda book: book.price, reverse=True)

def __iter__(self):
    return iter(self.books)

def __str__(self) -> str:
    return '\n'.join(str(book) for book in self.books)

```

Данный код реализует структуру данных «Массив», где элементами выступают экземпляры класса `Book`. Каждый экземпляр содержит данные о книге: автора, издательство, количество страниц, цену и ISBN. Основные компоненты программы — классы `Book` и `BookArray`, а также функции тестирования и бенчмарков `tests_array` и `benchmarks`.

Компоненты кода

1. Класс `Book`:

- Содержит данные о книге: `author`, `publisher`, `pages`, `price`, и `isbn`.
- Метод `__repr__` возвращает строковое представление объекта для удобного отображения книги.

2. Класс `BookArray`:

- `append`: Добавляет книгу (`Book`) в массив.
- `delete`: Удаляет книгу из массива по значению ISBN.
- `cocktail_sort_isbn`: Сортирует массив книг по возрастанию значения поля ISBN, используя алгоритм сортировки перемешиванием.
- `quick_sort_price`: Сортирует массив книг по убыванию значения поля `price` (цена) с помощью встроенной функции сортировки.
- `__iter__`: Делает массив итерируемым, что позволяет применять циклы для доступа к элементам.

Теперь проведем тесты и посмотрим на производительность

```

# Функция для тестирования класса BookArray
def tests_array() -> None:
    print("\nТесты:")
    arr = BookArray()

    # Тест 1. Проверка добавления элементов
    books = [
        Book("Джек Лондон", "Эксмо", 323, 699.99, "7-1343-16"),
        Book("Макс Фрай", "Эксмо", 687, 450.0, "4-4551-18"),
        Book("Владимир Торин", "Миф", 767, 899.0, "1-145-17"),
        Book("Джон Милтон", "АСТ", 445, 399.99, "6-4166-815"),
        Book("Уильям Индик", "Миф", 380, 500.0, "5-51-1998"),
        Book("Кэролайн О`Дохонью", "LikeBook", 365, 799.0, "2-190-10"),
        Book("Лия Арден", "Эксмо", 414, 1280.0, "8-21-3145"),
        Book("Михаил Булгаков", "Азбука", 1114, 1349.0, "9-5631-156"),

```

```

        Book("Гастон Леру", "Эксмо", 317, 320.0, "3-1342-2413"),
        Book("Бернар Вербер", "Эксмо", 412, 699.99, "8-3165-3614")
    ]
    print(f"Array до добавления в него элементов:\n{arr}")
    for book in books:
        arr.append(book)
    print(f"Array после добавления в него элементов:\n{arr}")

    # Тест 2. Проверка метода сортировки перемешиванием (по возрастанию ISBN)
    arr.cocktail_sort_isbn()
    print(f"\nArray после сортировки перемешиванием (по возрастанию ISBN):\n{arr}")

    # Тест 3. Проверка метода быстрой сортировки (по убыванию стоимости)
    arr.quick_sort_price()
    print(f"\nArray после быстрой сортировки (по убыванию стоимости):\n{arr}")

    # Тест 4. Удаление элементов по полю ISBN
    arr.delete("2-190-10")
    arr.delete("7-1343-16")
    arr.delete("1-145-17")
    arr.delete("8-3165-3614")
    arr.delete("4-4551-18")
    print(f"\nArray после удаления 5 элементов:\n{arr}\n")

    # Тест 5. Проверка итерируемости
    print("Проверка итерируемости:")
    for book in arr:
        print(book)

    print("Все тесты пройдены!")

# Функция для выполнения бенчмарков
def benchmarks() -> None:
    print("\nБенчмарки")
    arr = BookArray()

    # Создание 10,000 книг с уникальными значениями
    books = [Book(f'Author{i}', f'Publisher{i}', i + 100, i * 100.0, f'ISBN{i}')]
    for i in range(10000)]

    # Бенчмарк на добавление 10,000 элементов
    start = time.time()
    for book in books:
        arr.append(book)
    end = time.time()
    print(f"Время добавления 10,000 элементов: {(end - start):.5f} секунд")

    # Бенчмарк на сортировку перемешиванием
    start = time.time()
    arr.cocktail_sort_isbn()
    end = time.time()

```

```

    print(f"Время сортировки перемешиванием для 10,000 элементов: {(end -
start):.5f} секунд")

# Бенчмарк на сортировку по убыванию стоимости
start = time.time()
arr.quick_sort_price()
end = time.time()
print(f"Время быстрой сортировки для 10,000 элементов: {(end - start):.5f}
секунд")

# Запуск тестов и бенчмарков
tests_array()
benchmarks()

```

Результаты тестов:

Тесты:

Array до добавления в него элементов:

Array после добавления в него элементов:

Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')

Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')

Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')

Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')

Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')

Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')

Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')

Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')

Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')

Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')

Array после сортировки перемешиванием (по возрастанию ISBN):

Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')

Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')

Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')

Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')

```
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
```

Array после быстрой сортировки (по убыванию стоимости):

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Владимир Торин', publisher='Миф', pages=767, price=899.0, ISBN='1-145-17')
Book(author='Кэролайн О`Дохонью', publisher='LikeBook', pages=365, price=799.0, ISBN='2-190-10')
Book(author='Джек Лондон', publisher='Эксмо', pages=323, price=699.99, ISBN='7-1343-16')
Book(author='Бернар Вербер', publisher='Эксмо', pages=412, price=699.99, ISBN='8-3165-3614')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Макс Фрай', publisher='Эксмо', pages=687, price=450.0, ISBN='4-4551-18')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
```

Array после удаления 5 элементов:

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
```

Проверка итерируемости:

```
Book(author='Михаил Булгаков', publisher='Азбука', pages=1114, price=1349.0, ISBN='9-5631-156')
```

```
Book(author='Лия Арден', publisher='Эксмо', pages=414, price=1280.0, ISBN='8-21-3145')
Book(author='Уильям Индик', publisher='Миф', pages=380, price=500.0, ISBN='5-51-1998')
Book(author='Джон Мильтон', publisher='АСТ', pages=445, price=399.99, ISBN='6-4166-815')
Book(author='Гастон Леру', publisher='Эксмо', pages=317, price=320.0, ISBN='3-1342-2413')
Все тесты пройдены!

Бенчмарки
Время добавления 10,000 элементов: 0.00100 секунд
Время сортировки перемешиванием для 10,000 элементов: 2.84937 секунд
Время быстрой сортировки для 10,000 элементов: 0.00100 секунд
```

Можно заметить, что класс верно выполнил все функции, значит код работает корректно

4. Вывод

В ходе лабораторной работы была реализована структура данных «Массив», состоящий из объектов класса `Book`. В рамках работы были использованы три алгоритма сортировки: сортировка перемешиванием (cocktail sort), быстрая сортировка (quick sort) и сортировка выбором.

Сортировка перемешиванием (cocktail sort): данный алгоритм сортировки является двусторонним пузырьковым методом. Он одновременно проходит по массиву слева направо и справа налево, меняя элементы местами, если это необходимо. В отличие от обычной пузырьковой сортировки, этот алгоритм выполняет несколько проходов через массив, улучшая его эффективность, так как в процессе сортировки элементы перемещаются в обе стороны.

Быстрая сортировка (quick sort): данный алгоритм работает по принципу "разделяй и властвуй". Сначала выбирается опорный элемент, затем массив разделяется на две части: элементы, меньшие опорного, и элементы, большие опорного. После чего происходит рекурсивная сортировка обеих частей. Быстрая сортировка эффективно работает на больших объемах данных благодаря своей логарифмической сложности в среднем.

Сортировка выбором (selection sort): этот алгоритм начинает сортировку с первого элемента массива. На каждом шаге он находит минимальный (или максимальный) элемент в оставшейся неотсортированной части массива и меняет его местами с первым элементом неотсортированной части. Процесс продолжается до тех пор, пока не отсортирован весь массив. Сортировка выбором является простым, но не очень эффективным методом для больших наборов данных, так как она имеет квадратичную сложность.

Кроме того, в рамках лабораторной работы были реализованы функции для добавления, удаления и сортировки книг, а также выполнены тесты для проверки корректности работы этих операций и бенчмарки для измерения производительности алгоритмов на большом количестве данных.

В результате лабораторной работы была достигнута цель — создание структуры данных, обеспечивающей эффективное выполнение базовых операций с книгами, а также демонстрация применения и сравнение различных методов сортировки.