

Пользовательская документация

Описание продукта

Программа предназначена для сравнения производительности различных алгоритмов сортировки. Она позволяет:

- Загружать массив целых чисел из текстового файла.
- Выбирать алгоритмы сортировки для анализа.
- Сохранять результаты сравнения (время выполнения, использование памяти, количество сравнений и замен) в выходной файл.

Руководство по скачиванию

Скопируйте репозиторий:

git clone <https://github.com/Kirill-Dementev/Sorting-algorithms>

Этот этап находится на доработке, позже планируется сделать запуск программу через .exe файл (то есть что от пользователя не требовалась установленная java на ПК).

Использование программы

- 1) Создайте .txt файл (например: test.txt).
- 2) Введите в этот файл целые (в диапазоне от -2^{31} до $2^{31} - 1$) числа через пробел (например: 5 -3 10 0 42).
- 3) Запустите программу. Вы получите:
Введите путь к входному файлу:
- 4) Нужно ввести полный путь к вашему текстовому файлу (например: C:\Users\user\test.txt)
Далее программа выведет:
Выберите алгоритмы сортировки (цифры через пробел):
 - 1 - Пузырьковая сортировка (BubbleSort)
 - 2 - Быстрая сортировка (QuickSort)
 - 3 - Сортировка слиянием (MergeSort)
 - 4 - Сортировка вставками (InsertionSort)
 - 5 - Сортировка выбором (SelectionSort)
- 5) Выбираете алгоритмы (например, если вы хотите выбрать пузырьковую, быструю и слиянием сортировки, то напишите: 1 2 3), порядок выбора не важен.
Далее программа выведет:
Введите имя выходного файла:
- 6) Тут вы пишете название файла, который создаст программа и сохранит отчет. (например, test2.txt)

- 7) Потом программа выдаст: Отчёт записан в файл test2.txt.
Программа создаст файл в том каталоге, из которого вы запускаете программу.

Результат

По итогу в файле будет сохранен отчет, про каждый алгоритм будет написано: время работы, использованная память, количество сравнений элементов и количество замен местами элементов.

Пример:

Пузырьковая сортировка (BubbleSort):

Время выполнения - 13.170542801 секунд

Затраченная память - 0.0 Кб

Количество сравнений - 4999900545

Количество перемещений - 2502143107

Быстрая сортировка (QuickSort):

Время выполнения - 0.016639501 секунд

Затраченная память - 0.0 Кб

Количество сравнений - 360433

Количество перемещений - 277597

Сортировка слиянием (MergeSort):

Время выполнения - 0.015410801 секунд

Затраченная память - 8046.078125 Кб

Количество сравнений - 1536122

Количество перемещений - 1668928

Во всех алгоритмах, кроме сортировки слиянием пишется память 0 Кб, потому что они почти что не используют дополнительной памяти, а работают с исходным массивом.

Возможные ошибки

1) Не найден входной файл по пути <введенный вами путь>

2) Неверный тип данных в файле

Это ошибка означает, что в файле содержатся не только целые числа (возможно дробные числа или буквы и тд)

3) В файле содержится больше 100000 чисел

4) Неверный выбор алгоритма

Эта ошибка появляется, если при выборе алгоритма, вы вписали числа не в таком формате, как требуются, или содержатся числа не от 1 до 5, или вы ввели какие-то символы, а не только числа через пробел.

5) Ошибка записи в файл.

Техническая документация

Классы и интерфейсы в программе

- 1) Main: Главный класс. Взаимодействует с пользователем и использует другие классы. Содержит метод main (точка входа в программу).
- 2) InputOutputData: Класс для обработки входных данных и записывании данных в файл. Содержит два метода: readNumFile (считывает данные из файла и формирует массив, также обрабатывает случаи некорректного ввода), CreateOutputFile (создает отчет).
- 3) InvalidDataException: Класс, наследованный от класс Exception. Нужен для выбрасывания исключений.
- 4) InformationSort: Класс, хранящий информацию об алгоритме сортировке. То есть на основе этого класса будет создаваться объект для каждого алгоритма сортировки.
- 5) SortAlgorithm: Интерфейс для алгоритмов сортировки. То есть каждый алгоритм сортировки будет имплементироваться от этого интерфейса .
- 6) BubbleSort: Класс реализующий пузырьковую сортировку.
- 7) QuickSort: Класс реализующий быструю сортировку.
- 8) MergeSort: Класс реализующий сортировку слиянием.
- 9) InsertionSort: Класс реализующий сортировку вставками.
- 10) SelectionSort: Класс реализующий сортировку выбором.

Алгоритмы сортировки

Сортировка	Краткое описание	Худший случай	Средний случай	Лучший случай	Память
Пузырьком	Сравнение и обмен соседних элементов.	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$
Быстрая	Рекурсивное разделение массива.	$O(n^2)$	$O(n \log n)$	$O(n \log n)$	$O(\log n)$
Слиянием	Разделение и слияние подмассивов.	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Вставками	Постепенное построение отсортированной части.	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$
Выбором	Поиск минимального элемента и обмен.	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$

Измерение производительности

Время выполнения: замеряется с помощью `System.nanoTime()`.

Память: рассчитывается через `Runtime.getRuntime().totalMemory()` - `freeMemory()`.

Операции: счетчики `comparisons` и `swaps` в каждом алгоритме.

Требования

JDK17+, Git, IDE (в новой версии ожидается запуск через `.exe` файл, а значит эти требования будут не нужны).