Министерство Образования, Культуры и Исследований

Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Реферат**

Алгоритмы и структуры данных

На тему: “**Способы сортировки**”

**Выполнил:**

студент группы IA2102

Gavriusenco Stepan

Проверил преподаватель:

Mihail Croitor

Кишинев 2023

Оглавление

[**Введение:** 2](#_Toc130819107)

[**Типы сортировок:** 2](#_Toc130819108)

[**Алгоритм сортировки пузырьком:** 2](#_Toc130819109)

[**Алгоритм сортировки вставками:** 3](#_Toc130819110)

[**Быстрая сортировка:** 4](#_Toc130819111)

[**Сортировка слиянием:** 4](#_Toc130819112)

[**Поразрядная сортировка:** 5](#_Toc130819113)

[**Вывод:** 5](#_Toc130819114)

[**Библиография:** 6](#_Toc130819115)

# **Введение:**

Сортировка - это одна из наиболее часто используемых операций в программировании.

Сортировка - это процесс упорядочения набора данных в соответствии с определенным критерием, например, по возрастанию или убыванию.

# **Типы сортировок:**

Существует множество алгоритмов сортировки, но все они можно разделить на несколько типов.

**Простые алгоритмы**: сортировка пузырьком, сортировка вставками, сортировка выбором.

**Быстрые алгоритмы**: быстрая сортировка, сортировка слиянием.

**Распределительные алгоритмы**: поразрядная сортировка, сортировка подсчетом.

# **Алгоритм сортировки пузырьком:**

Сортировка пузырьком - это один из наиболее простых алгоритмов сортировки.

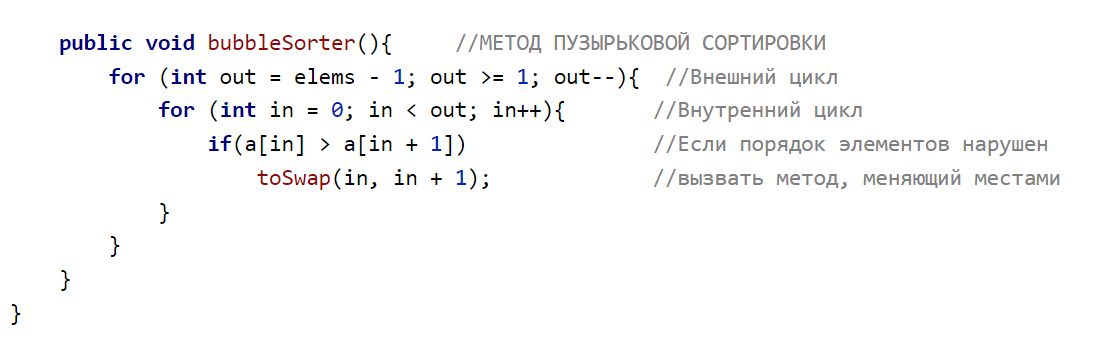
Алгоритм проходит по списку несколько раз, на каждом проходе сравнивая пары элементов и меняя их местами, если они находятся в неправильном порядке.

Теоретическая сложность: O(n^2)

Количество сравнений: O(n^2)

Количество перестановок: O(n^2)

Пример использования: сортировка списка имен по алфавиту.



# **Алгоритм сортировки вставками:**

Сортировка вставками - это алгоритм сортировки, в котором элементы массива просматриваются по одному и помещаются в соответствующее место в уже отсортированной части массива.

Теоретическая сложность: O(n^2) в худшем случае, O(n) в лучшем случае, когда массив уже отсортирован.

Количество сравнений: O(n^2) в худшем случае, O(n) в лучшем случае.

Количество перестановок: O(n^2) в худшем случае, O(n) в лучшем случае.

Пример использования: сортировка списка дат по возрастанию.

**public** **int**[] innerSort(**int**[] integers) {

**for** (**int** i = 1; i < integers.length; i++) {

**int** x = integers[i];

**int** j = i;

**while** (j > 0 && integers[j - 1] > x) {

integers[i] = integers[i - 1];

--j;

}

integers[j] = x;

}

**return** integers;

}

# **Быстрая сортировка:**

Быстрая сортировка - это один из наиболее эффективных алгоритмов сортировки.

Алгоритм основан на принципе "разделяй и властвуй", когда массив разбивается на подмассивы, которые затем сортируются отдельно.

Теоретическая сложность: O(n log n) в среднем случае, O(n^2) в худшем случае.

Количество сравнений: O(n log n) в среднем случае, O(n^2) в худшем случае.

Количество перестановок: O(n log n) в среднем случае, O(n^2) в худшем случае.

Пример использования: сортировка массива чисел.

**public** **int**[] innerSort(**int**[] array) {

**if** (array.length > 1) {

quickSortImpl(array, 0, array.length - 1);

}

**return** array;

}

**private** **int** **partition**(**int**[] array, **int** leftValueIndex, **int** reeghtValueIndex) {

**int** partitionValue = array[reeghtValueIndex];

**int** partitionIndex = leftValueIndex;

**for** (**int** i = leftValueIndex; i < reeghtValueIndex; i++) {

**if** (array[i] <= partitionValue) {

functions.swapValues(array, i, partitionIndex);

++partitionIndex;

}

}

functions.swapValues(array, partitionIndex, reeghtValueIndex);

**return** partitionIndex;

}

**private** **void** **quickSortImpl**(**int**[] array, **int** leftValue, **int** rightValue) {

**if** (leftValue < rightValue) {

**int** partitionIndex = partition(array, leftValue, rightValue);

quickSortImpl(array, leftValue, partitionIndex - 1);

quickSortImpl(array, partitionIndex + 1, rightValue);

}

}

# **Сортировка слиянием:**

Сортировка слиянием - это алгоритм сортировки, в котором массив разбивается на две половины, каждая из которых сортируется отдельно, а затем объединяется в отсортированный массив.

Теоретическая сложность: O(n log n) в худшем и лучшем случае.

Количество сравнений: O(n log n) в худшем и лучшем случае.

Количество перестановок: O(n log n) в худшем и лучшем случае.

Пример использования: сортировка списка клиентов по их фамилиям.

**private** **void** **sortImpl**(**int**[] integers, **int**[] arrayBuffer, **int** leftIndex, **int** rightIndex) {

**if** (leftIndex < rightIndex) {

**int** middle = (leftIndex + rightIndex) / 2;

sortImpl(integers, arrayBuffer, leftIndex, middle);

sortImpl(integers, arrayBuffer, middle + 1, rightIndex);

**int** k = leftIndex;

**for** (**int** i = leftIndex, j = middle + 1; i <= middle || j <= rightIndex; ) {

**if** (j > rightIndex || (i <= middle && integers[i] < integers[j])) {

arrayBuffer[k] = integers[i];

++i;

} **else** {

arrayBuffer[k] = integers[j];

++j;

}

++k;

}

**for** (**int** i = leftIndex; i <= rightIndex; i++) {

integers[i] = arrayBuffer[i];

}

}

}

# **Поразрядная сортировка:**

Поразрядная сортировка - это алгоритм сортировки, который сортирует элементы массива по отдельным разрядам.

Алгоритм используется для сортировки чисел или строк, где каждый элемент имеет одинаковую длину.

Теоретическая сложность: O(d \* (n + k)), где d - количество разрядов, n - количество элементов, k - количество возможных значений каждого разряда.

Количество сравнений: зависит от конкретной реализации.

Количество перестановок: зависит от конкретной реализации.

Пример использования: сортировка списка телефонных номеров по возрастанию.

# **Вывод:**

В своей лабораторной работе я сделал 3 метода сортировки (BubbleSort , InsertionSort, selectionSort).Работа представлена на Moodle, способом добавления ссылки на репозиторий на GitHub, для последующего оценивания**.**

# **Библиография:**

[**https://habr.com/ru/sandbox/165737/**](https://habr.com/ru/sandbox/165737/)

[**https://github.com/**](https://github.com/)

[**https://moodle.usm.md/mod/assign/view.php?id=231375**](https://moodle.usm.md/mod/assign/view.php?id=231375)