Создание компонента, реализующего

**алгоритм Сортировки пузырьком (Bubble sort)**

Выполнила:

Комарова Юлия

Группа 21ПИ2

1. Описание алгоритма

Сортировка пузырьком (Bubble sort) является наиболее простым для понимания и реализации алгоритмом сортировки массивов данных. Данный алгоритм многократно проходит через массив, меняя соседние элементы местами при необходимости. В процессе сортировки пузырьком на каждой итерации наибольший (или наименьший) элемент «всплывает» (или «опускается»). Так происходит до тех пор, пока массив не будет отсортирован.

Рассмотрим сортировку пузырьком по возрастанию. В процессе поочерёдного сравнения двух соседних элементов больший из двух будет вставать на «правильную» позицию путём перемены мест этих элементов, если это необходимо. После первого прохода по массиву наибольший элемент «всплывёт», то есть окажется на последнем месте в списке. При следующем проходе алгоритм пройдёт по всем элементам, кроме последнего, так как в процессе каждого прохода по массиву на своё место встаёт как минимум один элемент. Таким образом, количество сравниваемых элементов будет уменьшаться с каждым проходом. При реализации сортировки была использована оптимизация алгоритма с помощью проверки наличия обменов во время прохода. Если обменов не было, то следующего прохода по массиву не будет и сортировка завершится. Благодаря этой оптимизации значительно сокращается время работы сортировки в некоторых случаях.

1. Оценка сложности

Сложность сортировки пузырьком в среднем составляет O(n^2), где n - количество элементов в массиве. В лучшем случае сложность алгоритма будет равна O(n), так как гарантированно будет совершён как минимум один проход по массиву. В худшем случае, когда массив отсортирован в обратном порядке, сложность составит O(n^2).

1. Реализация сортировки пузырьком

Для реализации сортировки пузырьком был создан компонент на языке программирования C с использованием компонентов фреймворка Eco. Функция сортировки пузырьком имеет сигнатуру встроенной функции qsort и принимает на вход указатель на массив, количество элементов в массиве, размер одного элемента в байтах и функцию сравнения двух элементов такого же типа данных, что и массив. Это позволяет осуществлять сортировку на разных типах данных, используя одинаковую сигнатуру.

1. Тестирование

Сортировка тестировалась на разных типах данных: int, long, float, double, long double. Для каждого типа данных генерировались массивы случайных чисел размерами 1000, 5000, 10000 и 50000 элементов. Для каждого массива измерялось время его сортировки с помощью сортировки пузырьком и встроенной быстрой сортировки. Результаты тестирования продемонстрированы на графиках ниже. Графики построены на основе результатов измерений времени исполнения сортировок (эти результаты записывались в файл «results.csv»).

По мере увеличения размера входного массива данных время выполнения обеих сортировок растёт. Однако время выполнения быстрой сортировки растёт намного медленнее. Это связано с тем, что асимптотическая сложность быстрой сортировки в среднем составляет O(n log n), а сложность сортировки пузырьком - O(n^2). Предоставленные данные о времени работы сортировок демонтируют превосходство быстрой сортировки над сортировкой пузырьком.

1. Заключение

Итак, сортировка пузырьком является простым алгоритмом сортировки, который сравнительно хорошо подходит для небольших массивов данных или для частично отсортированных массивов. Однако для работы с большими объёмами данных рекомендуется использовать более эффективные алгоритмы сортировки, например, быструю сортировку.