Ивлев Андрей Б19-511

Лабораторная работа №1

По курсу «Языки программирования и методы программирования» (информатика, 3 семестр)

Техническое задание

Введение

На основе реализации класса Sequence<T>, полученной в рамках курса за 2-й семестр, была написана программа на C++, способная производить сортировку (достаточно произвольных) исходных данных с помощью одного из 4 алгоритмов сортировки, а также сравнивать скорость выполнения того или иного алгоритма.

Алгоритмы сортировки

В рамках лабораторной работы были реализованы следующие алгоритмы:

1. “Bubble Sort” (bubbleSort()). Сортировка пузырьком – простейший способ отсортировать исходные данные. Алгоритм проходит несколько раз по массиву, каждый раз меняя соседние элементы, если они стоят в неправильном порядке.
2. “Merge Sort” (mergeSort()). Сортировка слиянием, реализован вариант нисходящего слияния. Массив рекурсивно разбивается пополам до тех пор, пока его размер не становится равным 1, а затем полученные массивы последовательно соединяются друг с другом.
3. “Quick Sort” (quickSort()). Быстрая сортировка, за опорный элемент берется крайний левый элемент участка. Сначала массив разбивается на 3 непрерывные части: элементы, меньшие чем опорный, равные опорному и большие опорного. После чего рекурсивно сортируются участки с элементами меньшими, и большими опорного.
4. “Bogosort” (bogoSort()). Шуточная сортировка, которая проверяет отсортированы ли данные, и если нет, то в случайном порядке переставляет их до тех пор, пока данные не будут отсортированы.

Все алгоритмы сортировки принимают в качестве параметра способ сравнения элементов. В программе реализованы два способа сравнения: меньше или равно (asc()), больше или равно (desc()). Эти функции позволяют соответственно сортировать массивы по возрастанию или по убыванию.

Проверка работы алгоритмов

Все алгоритмы (кроме “Bogosort”) покрыты тестами, проверяющими их работу с последовательностями нулевой, единичной и произвольной длины, последовательностями, содержащими равные или различные значения.

Также присутствует консольный интерфейс, позволяющий проверить корректность работы алгоритмов на последовательностях, любой длины и содержания. Присутствует возможность создания произвольной последовательности фиксированной длины и отсортированной последовательности фиксированной длины. Исходная и отсортированная последовательность сохраняются в файле.

Консольное приложение также предоставляет возможность узнать время работы алгоритмов на тех или иных входных данных. В том числе возможно сравнение времени работы алгоритмов на одинаковых входных данных.