**Лабораторная работа №6**

**«Сортировка массивов»**

Сортировка

Сортировка это одна из наиболее востребованных задач. Прежде, примерно в конце 70-х, начале 80-х эта задача широко обсуждалась в литературе. Эта задача занимает значительную часть книг **Сэджвика «Фундаментальные алгоритмы на языке C++» и Кормена, Лейзерсона, Риверса «Алгоритмы. Построение и анализ»**. В наше время интерес к этой проблеме ослаб. Сортировка легко осуществляется при помощи стандартных программ. Но эта задача сохраняет своё значение, как полигон для отработки навыков программирования. Сегодняшняя наша задача рассмотреть три алгоритма сортировки и практически сравнить их эффективность.

Во всех сортировках одномерного массива организуется двойной цикл: внешний цикл отвечает за перебор всех элементов массива, а внутренний цикл – уже за то, как мы эти элементы будем сравнивать друг с другом и затем переставлять (или не переставлять).

Простейшие алгоритмы сортировки

Пусть у нас имеется некоторый массив чисел. Мы хотим упорядочить его по возрастанию. Эта задача имеет много решений. Сегодня мы рассмотрим три простых метода. Но сначала напишем отладочную программу и вспомогательную функцию распечатки массива. (**Demo.cpp)**.

Сортировка с выбором наименьшего элемента

В этом методе, на шаге мы выбираем наименьший элемент и ставим его в крайнюю позицию. Организуется двойной цикл. Внешний цикл движется от первого элемента к последнему, внутренний – просматривает элементы справа от текущего элемента, находит среди них минимальный, и ставит этот минимальный элемент на место текущего. Этот метод наиболее простой и понятный. Его и рекомендуется использовать, когда возникает необходимость отсортировать небольшое количество чисел. (**Ex1.cpp)**.

Сортировка методом вставками (погружения)

В данном методе каждый текущий элемент, как бы погружается в массив слева, так что слева от текущего элемента всегда остаётся отсортированный массив. В отличие от предыдущего метода, элементы справа от текущего элемента не рассматриваются, а внутренний цикл нужен для того, чтобы обеспечить «погружение» текущего элемента. Преимуществом этого метода является то, что для того, чтобы установить нужный элемент на своё место, требуется сместить целый диапазон. На этом можно сэкономить количество операций. (**Ex2.cpp)**.

Обменная сортировка (сортировка пузырьком)

Внешний цикл обеспечивает многократное повторение внутреннего цикла, вся сортировка сосредоточена во внутреннем цикле. В нём, сравниваются соседние элементы, и в случае нарушения порядка следования они меняются местами. В результате работы внутреннего цикла, гарантировано, что на шаге текущий наибольший элемент займёт своё место и следующий внутренний цикл можно закончить на одно на одну позицию раньше. (**Ex3.cpp)**.

Этот вариант сортировки можно улучшить, если учесть, что если мы прошли внутренний цикл и не произвели ни одной перестановки, значит массивы уже отсортирован. А также, следующий проход внутреннего цикла можно вести до того места, где был произведён последний обмен.

Есть библиотечная функция сортировки в С++ - qsort() (**Ex4.cpp)**.

Поиск в отсортированном массиве.

Одно из преимуществ, которое даёт сортировка – это возможность быстро найти элемент внутри отсортированного массива. Если мы ищем конкретное значение, то можем применить процедуру активного поиска. Напишите функцию активного поиска. Для активного поиска мы должны взять границы массива, выбрать элемент между границами, сравнить с искомым элементом. Если искомый элемент больше выбранного элемента, то перенести левую границу, если меньше, то правую, а если равен, то успешно закончить процесс поиска **(Ex5.cpp).**

**Задание**

Составить программу упорядочения первых N, N ≤12, элементов массива X:

- методом обменной сортировки (сортировка пузырьком)

- методом пузырька (метод вставками)

- методом прямого выбора (поиском минимального или максимального элемента)