# Практическое задание № 3

**Тема 3.2. Алгоритмы сортировки**

**Цель работы:** изучить основные алгоритмы поиска и сортировки; провести сравнительный анализ различных алгоритмов поиска и сортировки.

**Формулировка задания № 3**

1. Изучить следующие методы ***сортировки***:

* включение;
* выбор;
* обмен;
* сортировка Шелла;
* сортировка Хоара;
* пирамидальная сортировка.

1. Реализовать упомянутые выше методы. Проанализировать время, затрачиваемое на каждый из них при одинаковом количестве измерений (количестве элементов в массиве).
2. Изучить алгоритмы ***поиска***:

* в неупорядоченном массиве:
  + линейный;
  + быстрый линейный;
* в упорядоченном массиве:
* быстрый;
* бинарный;
* блочный.

4. Реализовать данные алгоритмы в одном файле в виде отдельных подпрограмм (функций).

5. Проанализировать, на какой итерации при разных алгоритмах поиска было найдено искомое число.

**Указания к выполнению работы**

При выполнении каждого задания необходимо написать программу на языке C++. Все алгоритмы сортировки следует реализовать в одном файле в виде отдельных подпрограмм (функций), которые сортируют один и тот же массив. Аналогично все алгоритмы поиска нужно реализовать в одном файле в виде отдельных функций. Для заполнения массива следует использовать генератор случайных чисел. С помощью функции ***clock()*** необходимоопределить время, затраченное на каждый алгоритм сортировки. Разработать и программно реализовать средство для проведения экспериментов по определению временных характеристик алгоритмов сортировки. Следует провести эксперименты по определению временных характеристик алгоритмов сортировки. Результаты экспериментов представить в виде таблицы, ячейки которой содержат время выполнения алгоритма сортировки массива с заданным количеством элементов. Также необходимо провести эксперимент для упорядоченных, неупорядоченных и упорядоченных в обратном порядке массивов (для каждого типа массива заполнить отдельную таблицу). Кроме того, построить графики функций временной сложности алгоритмов сортировки.

Выполнение задания необходимо проводить в следующей последовательности:

* + - разработать графическую схему алгоритмов;
    - записать алгоритмы на языке C++;
    - разработать контрольный тест к программе;
    - отладить программу;
    - представить отчет по работе.

**Требования к отчету**

Структура отчета должна соответствовать приведённой выше последовательности и содержать:

* титульный лист;
* алгоритм решения задачи (схема алгоритма выполняется по ЕСПД (ГОСТ 19.003-80 и ГОСТ 19.002-80));
* листинг программы;
* контрольный тест;
* выводы.