**Лабораторная работа**

**ИССЛЕДОВАНИЕ И ОЦЕНКА АЛГОРИТМОВ СОРТИРОВКИ**

**Цель работы**. Разработка программ, реализующих различные рекурсивные алгоритмы, и оценка их временной и пространственной сложности.

**Задание.** Разработка программ, реализующих различные алгоритмы сортировки, и оценка их временной и пространственной сложности.

**Пузырьковая сортировка:**

Этот метод - наиболее распространенный и простой. Он требует минимального объема памяти для данных, но затраты времени на его реализацию велики. При упорядочении выполняются следующие операции:

1) элементы массива сравниваются *попарно*: первое со вторым; второе с третьим; *i*-тое – с (*i*+1) - вым;

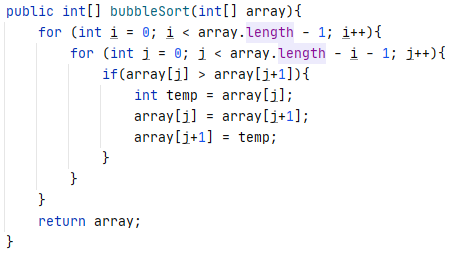
2) если они стоят неправильно (при *упорядочении* по возрастанию первый должен быть меньше второго или равен ему), то элементы меняются местами.

**Сортировка вставками:**

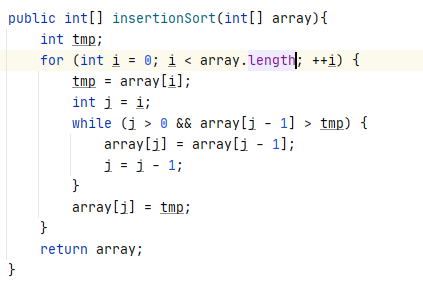
Это достаточно простой, но эффективный в некоторых случаях метод. В начале считается, что первый элемент находится на своем месте. Далее, начиная со второго, каждый элемент сравнивается со всеми, стоящими перед ним, и если они стоят неправильно, то меняются местами. Таким образом, новый элемент сравнивается и меняется местами не со всем массивом, а только до тех пор, пока в начале не найдется элемент, меньший его. Поэтому рассматриваемый алгоритм примерно в два раза быстрее сортировки пузырьком. Для уже частично отсортированных массивов такой метод является наилучшим.

**Код программы.**

**Пузырьковая сортировка**



**Сортировка вставками**



**Формулы верхней оценки временной и емкостной сложности заданных алгоритмов.**

**Сортировка пузырьком**

Верхняя оценка сложности алгоритма: О(*n^2*)

Ёмкостная сложность алгоритма: О(1)

**Сортировка вставками**

Верхняя оценка сложности алгоритма: О(*n^2*)

Ёмкостная сложность алгоритма: О(1)

**Результаты экспериментальной оценки временной и емкостной сложности заданного алгоритма**.

**Bubble sort**

int i = 4 bytes

int j = 4 bytes

int temp = 4 bytes

4 + 4 + 4 = 12 => O(1)

**Insertion sort**

int i = 4 bytes

int j = 4 bytes

int temp = 4 bytes

4 + 4 + 4 = 12 => O(1)