**Сортировки**

Сортировки бывают внутренние и внешние.

Общеизвестны три элементарных методов сортировки: метод пузырька, сортировка выбором, и сортировка вставками. Если элементарная сортировка работает не медленней других частей программы, осуществляющей обработку данных (считывание данных или их вывод на печать, небольшой размер сортируемого массива ), то отпадает необходимость в поиске более быстрых способов сортировки.

Для сортировки случайно упорядоченного массива из N элементов, с применением элементарных методов, требуется время, пропорциональное . Это время может оказаться вполне приемлемо.

Элементарные сортировки не годятся для массивов больших размеров с произвольной организацией. Для этого может послужить метод сортировки Шелла, для которого при большом N требуется гораздо меньше чем шагов.

Внешняя и внутренняя сортировки. Рассмотрим только внутренние.

1. Обменная сортировка

- метод пузырька сложность метода ***O()***

-шейкерная сортировка сложность метода ***O()***

- обменная сортировка слиянием сложность метода ***O(N ˑ log(N))***

1. Сортировка посредством выбора

- метод простого выбора сложность метода ***O()***

- метод квадратичного выбора

1. Сортировка вставками

- метод простых вставок сложность метода составит ***O()***

- метод Шелла

**метод квадратичного выбора**

Рассмотрим 16 чисел, будем сортировать по убыванию. Разобьем их на 4 группы по 4 числа и найдем наибольший элемент в каждой подгруппе. Максимальный из них, и будет наибольшим элементом всей последовательности.

Для получения второго по величине элемента, рассмотрим 3 оставшихся из предыдущих максимумов. Плюс максимум из оставшихся трех элементов, из группы, содержащей наибольший. И т.д…….

Разбиваем массив на 4 группы

503 87 512 61 908 170 897 275 653 426 154 509 612 677 765 70

Начинаем с определения максимумов каждой группы

512 908 653 765

Тогда наибольший из них и будет наибольшим для всей последовательности. Это 908.

503 87 512 61 908 170 897 275 653 426 154 509 612 677 765 70

Для получения второго по величине элемента – рассмотрим оставшиеся три + наибольший из оставшихся в группе, содержащей 908 (170 897 275 ), т.е

512  897 653 765

503 87 512 61 908 170 897 275 653 426 154 509 612 677 765 70

Для получения третьего рассмотрим оставшиеся три + наибольший из оставшихся в группе, содержащей 897 (170 275 ), т.е. 512  275 653 765 и т.д…

сложность метода составит ***O(N)***

**метод Шелла**

делим на 2 количество элементов, получаем шаг

получилось 2 массива и методом простой вставки их сортируем

еще раз шаг делим на 2

получилось 2 массива и опять сортируем методом простой вставки

до тех пор пока шаг не станет = 1

503 87 512 61 908 170 897 275 653 426 154 509 612 677 765 703

503 87 154 61 612 170 765 275 653 426 512 509 908 677 897 703

503 87 154 61 612 170 512 275 653 426 765 509 908 677 897 703

154 61 503 87 512 170 612 275 653 426 765 509 897 67 908 703

61 87 154 170 275 426 503 509 512 612 653 677 703 765 897 908

первоначально используемая Шеллом последовательность длин промежутков:

***d 1 = N / 2 , d i = d i − 1 / 2 , d k = 1 {\displaystyle d\_{1}=N/2,d\_{i}=d\_{i-1}/2,d\_{k}=1} =N/2, = /2, …, = 1***

в худшем случае сложность алгоритма составит ***O()***

предложенная [Седжвиком](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%B4%D0%B6%D0%B2%D0%B8%D0%BA,_%D0%A0%D0%BE%D0%B1%D0%B5%D1%80%D1%82) последовательность: d i = 9 ⋅ 2 i − 9 ⋅ 2 i / 2 + 1 {\displaystyle d\_{i}=9\cdot 2^{i}-9\cdot 2^{i/2}+1}

**= 9 ˑ - 9 ˑ + 1** если i четное и d i = 8 ⋅ 2 i − 6 ⋅ 2 ( i + 1 ) / 2 + 1 {\displaystyle d\_{i}=8\cdot 2^{i}-6\cdot 2^{(i+1)/2}+1},

**= 8 ˑ - 6 ˑ + 1** если i нечетное.

сложность алгоритма при таких приращениях составит ***O()***

а в худшем случае сложность алгоритма составит ***O()***