**СОРТИРОВКА ПУЗЫРЬКОМ** - **BUBBLE SORT**

**Сортировка пузырьком** **(англ. bubble sort)**, сортировка простыми обменами, метод сортировки обменами — один из алгоритмов сортировки. По сравнению с другими алгоритмами считается простейшим для понимания и реализации. Эффективен для массивов небольшого размера.

n — размер массива, количество элементов массива. Сложность алгоритма: **O(n^2).**

Алгоритм считается учебным, на практике (вне учебной литературы) не применяется (на практике применяются более эффективные (совершенные) алгоритмы). Лежит в основе некоторых более эффективных алгоритмов, например, алгоритма **шейкерной сортировки**, алгоритма **пирамидальной сортировки** и алгоритма **быстрой сортировки**.

[O](https://ru.wikipedia.org/wiki/O_%D0%B1%D0%BE%D0%BB%D1%8C%D1%88%D0%BE%D0%B5)(n2)

**Алгоритм:**

Выполняется некоторое количество проходов по массиву — начиная от начала массива, перебираются пары соседних элементов массива. Если 1-й элемент пары больше 2-го, элементы переставляются (выполняется обмен).

Пары элементов массива перебираются (проходы по массиву повторяются) либо (n-1) раз, либо до тех пор, пока на очередном проходе не обнаружится, что более не требуется выполнять перестановки (обмены) (массив отсортирован).

При каждом проходе алгоритма по внутреннему циклу очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива (как бы «всплывает» до нужной позиции, как пузырёк в воде — откуда и название алгоритма.

**СОРТИРОВКА СЛИЯНИЕМ - MERGE SORT**

**Сортировка слиянием** **(англ. merge sort)** — алгоритм сортировки, который упорядочивает списки (или другие структуры данных, доступ к элементам, которых можно получать только последовательно, например — потоки, массивы) в определённом порядке.

Эта сортировка — хороший пример использования принципа «разделяй и властвуй». Сначала задача разбивается на несколько подзадач меньшего размера. Затем эти задачи решаются с помощью рекурсивного вызова или непосредственно, если их размер достаточно мал. Наконец, их решения комбинируются, и получается решение исходной задачи.

Алгоритм был изобретён **Джоном фон Нейманом** в 1945 году.

**Алгоритм:**

Для решения задачи сортировки эти три этапа выглядят так:

1. Сортируемый массив разбивается на две части примерно одинакового размера;
2. Каждая из получившихся частей сортируется отдельно, например — тем же самым алгоритмом;
3. Два упорядоченных массива половинного размера соединяются в один.

Рекурсивное разбиение задачи на меньшие происходит до тех пор, пока размер массива не достигнет единицы (любой массив длины 1 можно считать упорядоченным).

*Соединение двух упорядоченных массивов в один.*

Основную идею слияния двух отсортированных массивов можно объяснить на следующем примере. Пусть мы имеем два уже отсортированных по возрастанию подмассива. Тогда:

На каждом шаге мы берём меньший из двух первых элементов подмассивов и записываем его в результирующий массив. Счётчики номеров элементов результирующего массива и подмассива, из которого был взят элемент, увеличиваем на 1.

Когда один из подмассивов закончился, мы добавляем все оставшиеся элементы второго подмассива в результирующий массив.

Сложность алгоритма: **O(nlogn).**

**БЫСТРАЯ СОРТИРОВКА – СОРТИРОВКА ХОАРА – QUICKSORT**

**Быстрая сортировка, сортировка Хоара (англ. quicksort)**, часто называемая **qsort** (по имени в стандартной библиотеке языка Си) — алгоритм сортировки, разработанный английским информатиком (Сэр **Чарльзом Э́нтони Ри́чард Хо́ар)** во время его работы в **МГУ в 1960 году**.

Один из самых быстрых известных универсальных алгоритмов сортировки массивов: в среднем **O(nlogn)** обменов при упорядочении n элементов;

Из-за наличия ряда недостатков на практике обычно используется с некоторыми доработками.

**QuickSort** является существенно улучшенным вариантом алгоритма сортировки с помощью прямого обмена (его варианты известны как «**Пузырьковая сортировка**» и «**Шейкерная сортировка**»).

Принципиальное отличие состоит в том, что в первую очередь производятся перестановки на наибольшем возможном расстоянии и после каждого прохода элементы делятся на две независимые группы (таким образом улучшение самого неэффективного прямого метода сортировки дало в результате один из наиболее эффективных улучшенных методов).

Общая идея алгоритма состоит в следующем:

Выбрать из массива элемент, называемый опорным. Это может быть любой из элементов массива. От выбора опорного элемента не зависит корректность алгоритма, но в отдельных случаях может сильно зависеть его эффективность.

Сравнить все остальные элементы с опорным и переставить их в массиве так, чтобы разбить массив на три непрерывных отрезка, следующих друг за другом: «элементы меньшие опорного», «равные» и «большие».

Для отрезков «меньших» и «больших» значений выполнить рекурсивно ту же последовательность операций, если длина отрезка больше единицы.

На практике массив обычно делят не на три, а на две части: например, «меньшие опорного» и «равные и большие»; такой подход в общем случае эффективнее, так как упрощает алгоритм разделения.

*Хоар разработал этот метод применительно к машинному переводу; словарь хранился на магнитной ленте, и сортировка слов обрабатываемого текста позволяла получить их переводы за один прогон ленты, без перемотки её назад. Алгоритм был придуман Хоаром во время его пребывания в Советском Союзе, где он обучался в Московском университете компьютерному переводу и занимался разработкой русско-английского разговорника.*

**Алгоритм:**

Алгоритм состоит из трёх шагов:

1. Выбрать элемент из массива. Назовём его опорным.
2. Разбиение: перераспределение элементов в массиве таким образом, что элементы, меньшие опорного, помещаются перед ним, а большие или равные — после.
3. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один элемент или отсутствуют элементы.

Эффективность сортировки может сильно зависеть от способа выбора среднего элемента. Существует достаточно большое количество подходов, например: **Разбиение Ломуто, Схема Хоара.**

Для улучшения производительности при большом количестве одинаковых элементов в массиве может быть применена процедура разбиения массива на три группы: элементы меньшие опорного, равные ему и больше него (Бентли и Макилрой называют это «толстым разбиением». Данное разбиение используется в функции qsort в седьмой версии Unix)

Быстрая сортировка используется в стандарте языка C++, в частности, в методе sort структуры данных list.