

## QUICK SORT.

**Быстрая сортировка** представляет собой усовершенствованный метод сортировки, основанный на принципе обмена. Пузырьковая сортировка является самой неэффективной из всех алгоритмов прямой сортировки. Однако усовершенствованный алгоритм является лучшим из известных методом сортировки массивов. Он обладает столь блестящими характеристиками, что его изобретатель Ч. Хоар назвал его быстрой сортировкой. Для достижения наибольшей эффективности желательно производить обмен элементов на больших расстояниях. В массиве выбирается некоторый элемент, называемый **разрешающим**. Затем он помещается в то место массива, где ему полагается быть после упорядочивания всех элементов. В процессе отыскания подходящего места для разрешающего элемента производятся перестановки элементов так, что слева от них находятся элементы, меньшие разрешающего, и справа — большие (предполагается, что массив сортируется по возрастанию). Тем самым массив разбивается на две части:

- не отсортированные элементы слева от разрешающего элемента;
- не отсортированные элементы справа от разрешающего элемента.

Чтобы отсортировать эти два меньших подмассива, алгоритм рекурсивно вызывает сам себя.

Если требуется сортировать больше одного элемента, то нужно:

- выбрать в массиве разрешающий элемент;
- переупорядочить массив, помещая элемент на его окончательное место;
- отсортировать рекурсивно элементы слева от разрешающего;
- отсортировать рекурсивно элементы справа от разрешающего.

Ключевым элементом быстрой сортировки является **алгоритм переупорядочения**.

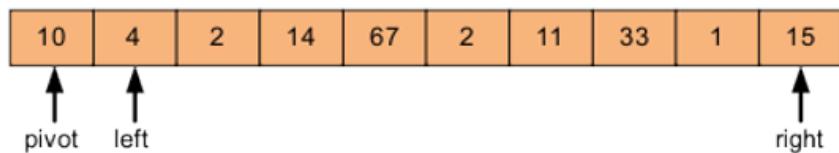
Рассмотрим сортировку на примере массива:

10, 4, 2, 14, 67, 2, 11, 33, 1, 15.

Для реализации алгоритма переупорядочения используем указатель `left` на крайний левый элемент массива. Указатель движется вправо, пока элементы, на которые он показывает, остаются меньше разрешающего. Указатель `right` поставим на крайний правый элемент массива, и он движется влево, пока элементы, на которые он показывает, остаются больше разрешающего.

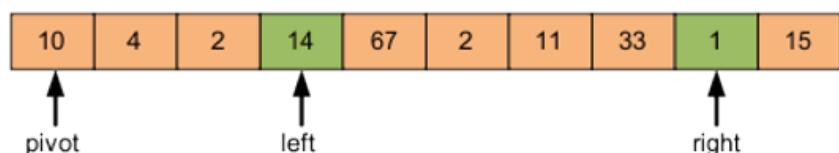
Пусть крайний левый элемент — разрешающий. Установим на следующий за ним элемент `right` — на последний. Алгоритм должен определить правильное

положение элемента 10 и по ходу дела поменять местами неправильно расположенные элементы.

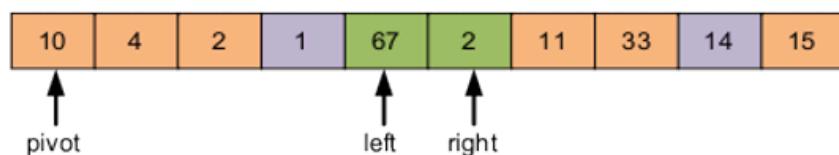


Движение указателей останавливается, как только встречаются элементы, порядок расположения которых относительно разрешающего элемента неправильный.

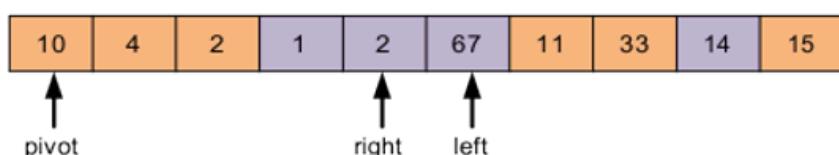
Указатель `left` перемещается до тех пор, пока не покажет элемент больше 10; `right` движется, пока не покажет элемент меньше 10.



Эти элементы меняются местами и движение указателей возобновляется.

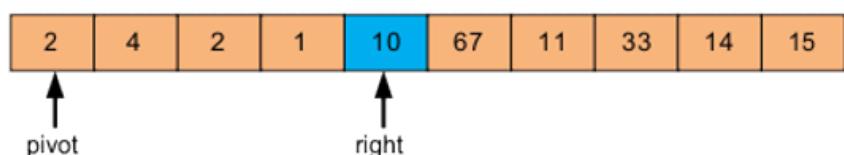


Процесс продолжается до тех пор, пока `right` не окажется слева от `left`.



Тем самым будет определено правильное место разрешающего элемента.

Осуществляется перестановка разрешающего элемента с элементом, на который указывает.



Разрешающий элемент находится в нужном месте: элементы слева от него имеют меньшие значения; справа — большие. Алгоритм рекурсивно вызывается для сортировки подмассивов слева от разрешающего и справа от него.