

Попробуем разбить исходный массив на два подмассива, относительно центрального элемента (назовём его медианой). Сделаем это следующим образом: если текущий элемент больше медианы, то перенесем его вправо иначе влево. По завершении этого процесса массив окажется немного более упорядоченным. Продолжим упорядочение, разбив каждый из уже обработанных подмассивов, ещё на два подмассива и проведём над ними ту же самую операцию. Полностью процесс сортировки окажется завершённым, когда длина подмассивов окажется единичной. Эта сортировка имеет один недостаток. Например, если исходный массив будет изначально полностью или почти полностью упорядоченным, это не уменьшит количество выполняемых операций. Быстрая сортировка всё равно проведет дробление исходного массива до победного конца.

Практически очевидно, что алгоритм сортировки разумно построить в виде рекурсивной процедуры, которая на вход получала бы отрезок исходного массива (в виде номеров элементов массива являющихся границами отрезка) и выполняла следующие операции:

- Выбор элемента медианы (можно середину)
- Для всех элементов слева от медианы
  - Если значение элемента больше значения – медианы, то переносим его вправо.
- Для всех элементов справа от медианы
  - Если значение элемента меньше значения – медианы, то переносим его влево.

### *Алгоритм*

- ОТРЕЗОК = весь исходный массив
- Вызов процедуры БЫСТРАЯ СОРТИРОВКА

Процедура БЫСТРАЯ СОРТИРОВКА (ЛЕВАЯ ГРАНИЦА, ПРАВАЯ ГРАНИЦА)

- Если ЛЕВАЯ меньше ПРАВОЙ то
  - Определяем МЕДИАНУ как середину между левой и правой
  - Для всех элементов между ЛЕВОЙ и МЕДИАННОЙ делать
    - Если ОЧЕРЕДНОЙ элемент больше МЕДИАНЫ то ОЧЕРЕДНОЙ перекинуть вправо от МЕДИАНЫ.
  - Для всех элементов между ПРАВОЙ и МЕДИАННОЙ делать
    - Если ОЧЕРЕДНОЙ элемент меньше МЕДИАНЫ то ОЧЕРЕДНОЙ перекинуть влево от МЕДИАНЫ.
  - БЫСТРАЯ СОРТИРОВКА(ЛЕВАЯ, МЕДИАНА)
  - БЫСТРАЯ СОРТИРОВКА(МЕДИАНА, ПРАВАЯ)