



# Сортировка Хоара (quicksort) на массивах

**Автор:** Соколов Егор Михайлович, 143 группа  
**Научный руководитель:** ст.пр. С.В. Григорьев

Санкт-Петербургский государственный университет

17 декабря 2015г.

- Сортировка Хоара, быстрая сортировка – алгоритм сортировки, разработанный информатиком Чарльзом Хоаром во время его работы в МГУ в 1960 году
- QuickSort считается одним из самых быстрых известных алгоритмов сортировки массивов: в среднем  $O(n \log n)$

# Существующие альтернативные алгоритмы

- Сортировка пузырьком
  - ▶ Среднее время  $O(n^2)$
  - ▶ Эффективен для небольших массивов
- Сортировка слиянием
  - ▶ На отсортированных массивах работает так же, как и на случайных
  - ▶ Требуется дополнительной памяти по размеру исходного массива
- Сортировка Шелла
  - ▶ Во многих случаях работает медленнее сортировки Хоара

**Целью** работы является исследование алгоритма Хоара  
**Задачи:**

- Исследовать алгоритм Хоара
- Реализовать алгоритм для массивов на языке F#
- Провести тестирование программы

- Выбирается элемент, называемый опорным
- Все элементы сравниваются с опорным, а затем переставляются в соответствии с правилом: «Меньше опорного» и «Больше опорного или равно»
- Выполняется рекурсивно та же последовательность операций для отрезков «Меньших» и «Больших»

# Достоинства и недостатки алгоритма

- Скорость работы – алгоритм в среднем работает за  $O(n \log n)$
- В текущей реализации – не требует дополнительной памяти в виде массива
- Алгоритм неустойчив

# Тестирование

- ..... ✓ return [], empty
- ..... ✓ return [0], zero
- ..... ✓ return [1; 2; 3; 4; 5], sorted
- ..... ✓ return [100; 50; 3; 1; 0], unsorted
- ..... ✓ return [5; 3; 2; 5; 1], not sorted

```
let partition (a:array<int>) l r piv =  
    let pivot = a.[piv]  
    swap piv r a  
    let mutable st = l  
    for i = l to r - 1 do  
        if a.[i] <= pivot  
        then  
            swap i st a  
            st <- st + 1  
    swap st r a  
    st  
  
let quickSort (a:array<int>) =  
    let rec loop (a:array<int>) l r =  
        if l < r  
        then  
            let piv = (l + r) / 2  
            let pivotN = partition a l r piv  
            loop a l (pivotN - 1)  
            loop a (pivotN + 1) r  
    loop a 0 (a.Length - 1)  
    a
```

- Исследована работа алгоритма Хоара
- Реализован алгоритм на ЯП F#
- Протестирован алгоритм с помощью системы NUnit