Algorithms and data structures

lecture #5. Quick sort

Mentor: <….>

lecture #5. Quick sort

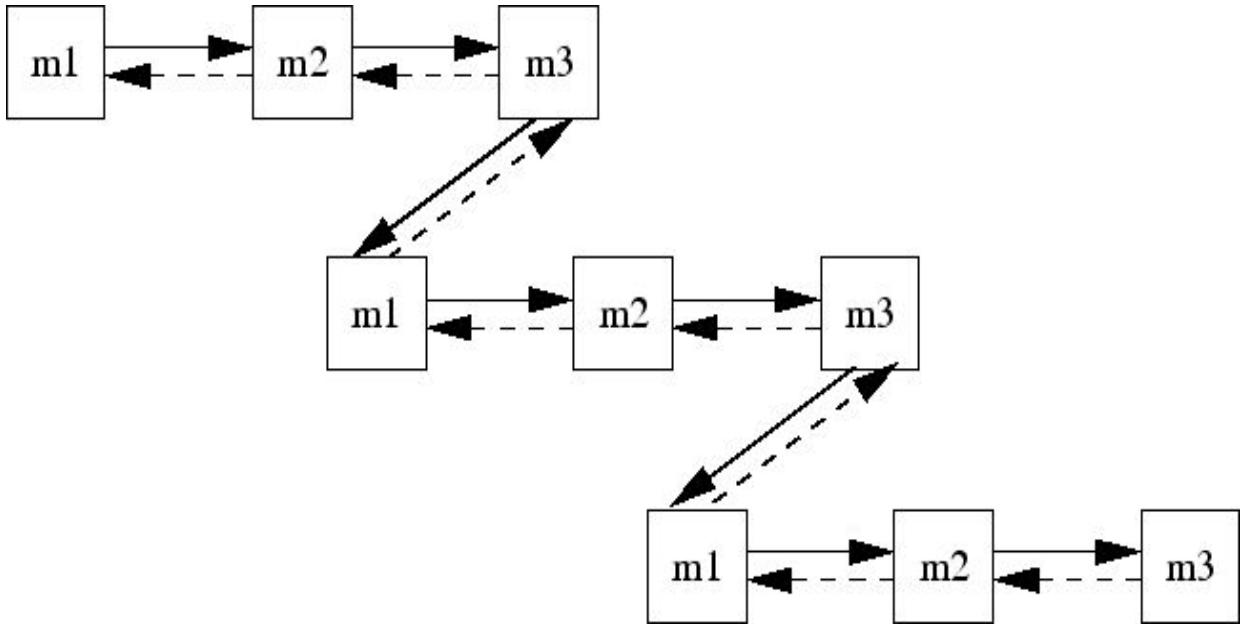
- Quick sort
  - Общая информация
  - Алгоритм разделения
  - Псевдокод
  - Детальный разбор на картинках
  - Реализация Java

- Indirect recursion (косвенная рекурсия)
  - Определение
  - Натуральный числа (прямая рекурсия)
  - Натуральный числа (косвенная рекурсия)

Косвенная рекурсия

Когда метод m1 вызывает другой метод m2, который вызывает m3, и m3 в свою очередь, вызывает исходный вызывающий метод m1.

- Основное отличие заключается в том, что косвенная рекурсия использует более одного метода.
- Программа обхода каталогов.



Натуральный числа (прямая рекурсия)

printNaturalNumbers(lower, upper)

if lower > upper -> base case
    return

  print(lowerRange)

printNaturalNumbers(lowerRange + 1, upperRange) -> recursive case

Натуральный числа (косвенная рекурсия)

```
printNaturalNumbers(lower, upper)
        if lower <= upper
                print(lower)
                lower += 1
                helperFunction(lower, upper)
        else return

helperFunction(lower, upper)
                if lower <= upper
                        print(lower)
                        lower += 1
                        printNaturalNumbers(lower, upper)
                else return
```

Quick sort (быстрая сортировка)

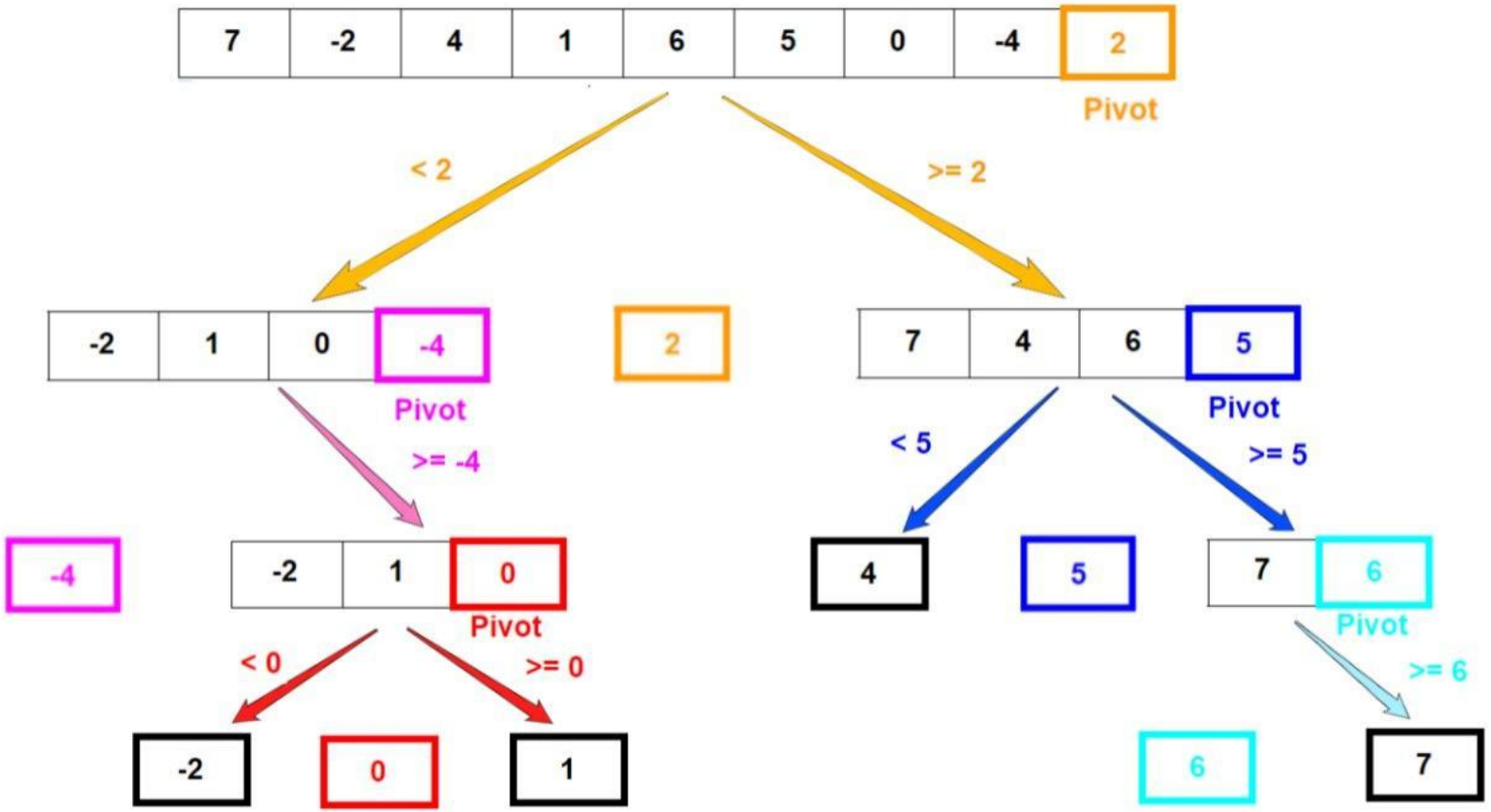
1.Выбираем опорный элемент из массива. Опорный элемент может быть любым в массиве.

2.Делим массив на 2 подмассива. Элементы, которые меньше опорного, и элементы, которые больше опорного.

3.Рекурсивно применяем сортировку к обоим подмассивам.

В результате массивы будут делиться до тех пор, пока не останется один элемент, который потом отсортируется.

Quick sort (быстрая сортировка)



https://dev-gang.ru/article/bystraja-sortirovka-v-javascript-964jzhnwc1/

Quick sort (быстрая сортировка)

```
quickSort(array[], startIndex, endIndex)

    if (startIndex >= endIndex){-> базовое условие
       return;
       }

        pivotIndex = helperFunction(array, startIndex, endIndex) -> новая опора

        quickSort(array, startIndex, pivotIndex – 1) -> все что слева

        quickSort(array, pivotIndex + 1, endIndex) -> все что справа

helperFunction (array[], low, high)
      pivotIndex = array[high];

      index = (low – 1)

      for (j = low; j <= high- 1; j++){
            if (arr[j] < pivot)
              index++
              swap arr[index] and arr[j]

    swap arr[index + 1] and arr[high])
    return (index + 1)
```