Сортировка массива

Сортировка - упорядочение некоторого множества элементов

$$a_0,\ a_1,\ a_2,\ \dots,\ a_{n-1} \quad \rightarrow \quad a'_0,\ a'_1,\ a'_2,\ \dots,\ a'_{n-1}$$

так, чтобы выполнялось

$$a'_0 \le a'_1 \le a'_2 \le \dots \le a'_{n-1}$$

Для числовых элементов обычно упорядочивают по возрастанию или убыванию

Алгоритм сортировки включает:

1) Правило, которое определяет упорядоченность пары элементов;

- *Ключ сортировки* это атрибут (или несколько атрибутов), по значению которого определяется порядок элементов. Для простых числовых данных это само число.
- *Правило сравнения* (операции отношения > и <)

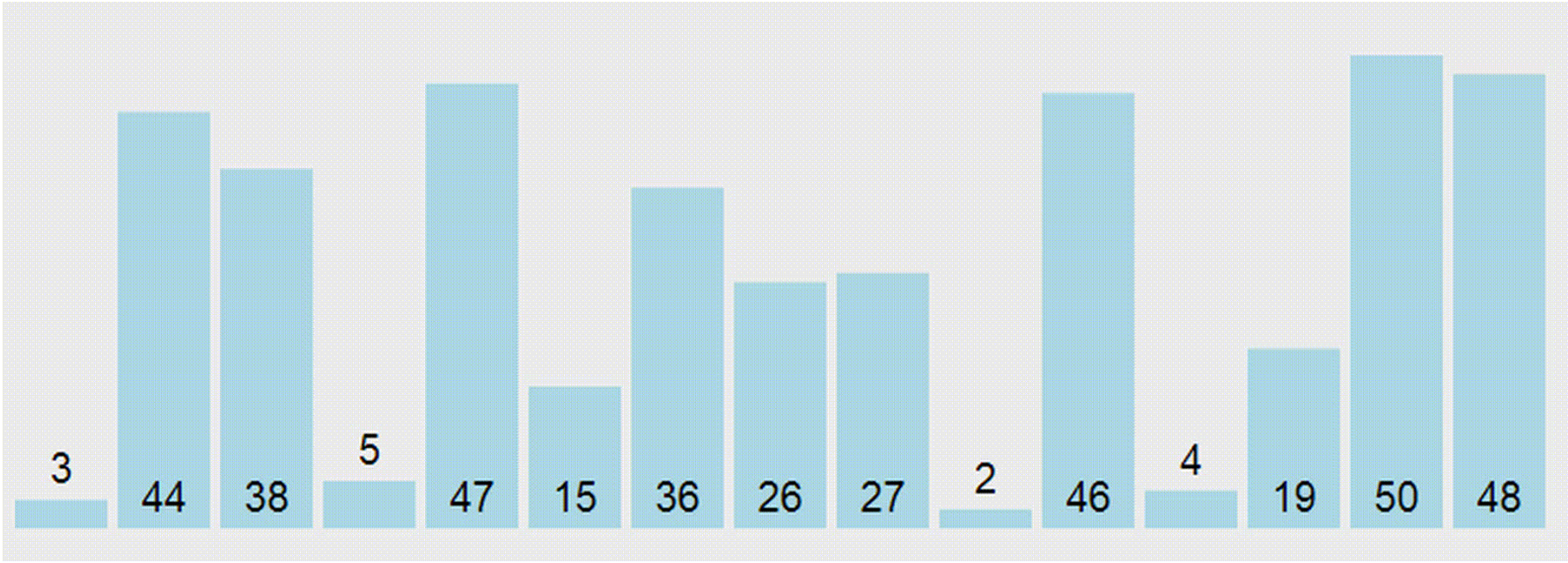
2) Порядок сортировки: возрастание, убывание;

3) Собственно сортирующий алгоритм, который осуществляет сравнение и перестановку элементов до тех пор, пока все элементы не будут упорядочены.

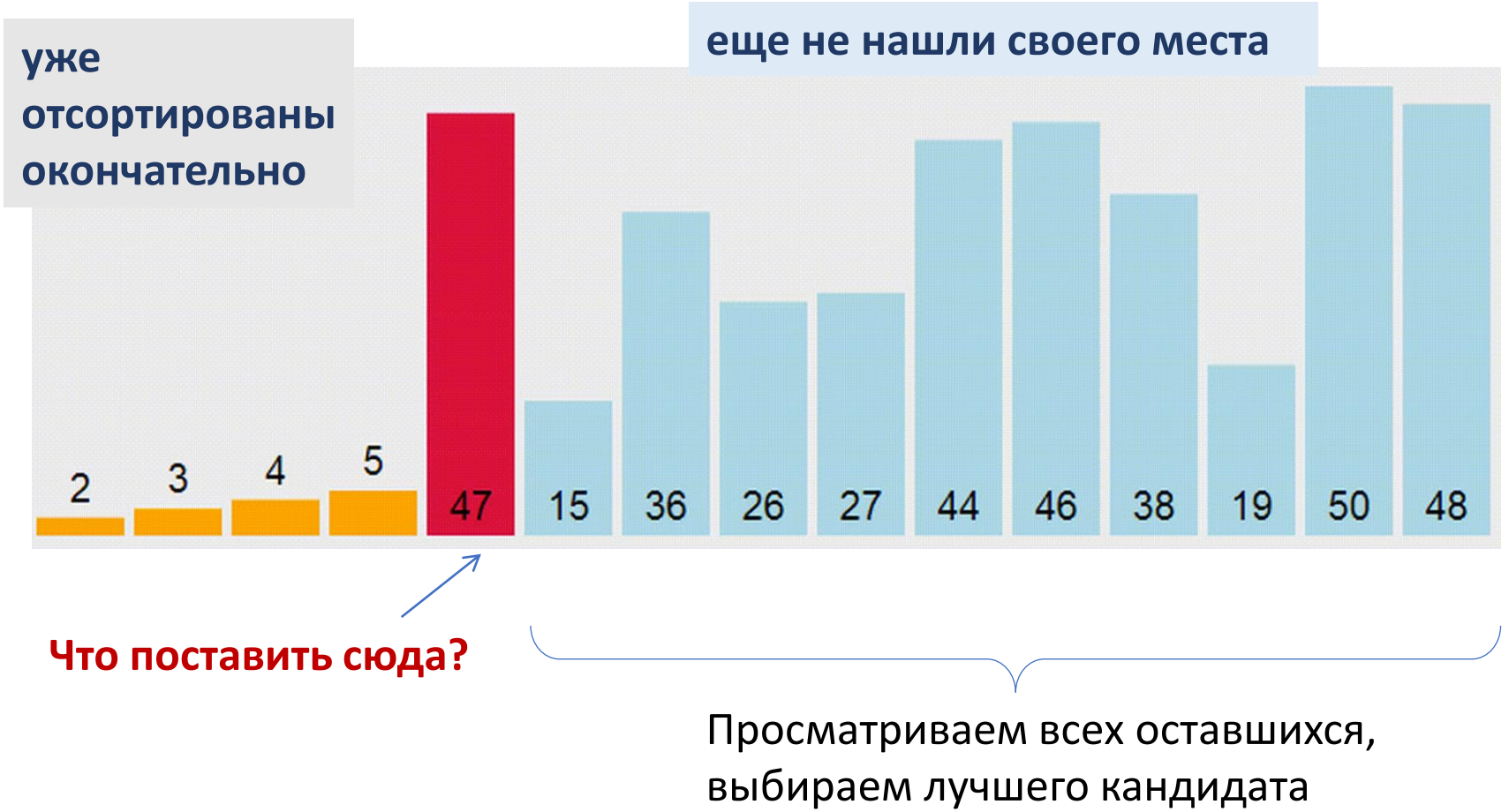
Простые сортировки:

- **Сортировка поиском**
- **Сортировка вставками**
- **Сортировка пузырьком**

Сортировка поиском

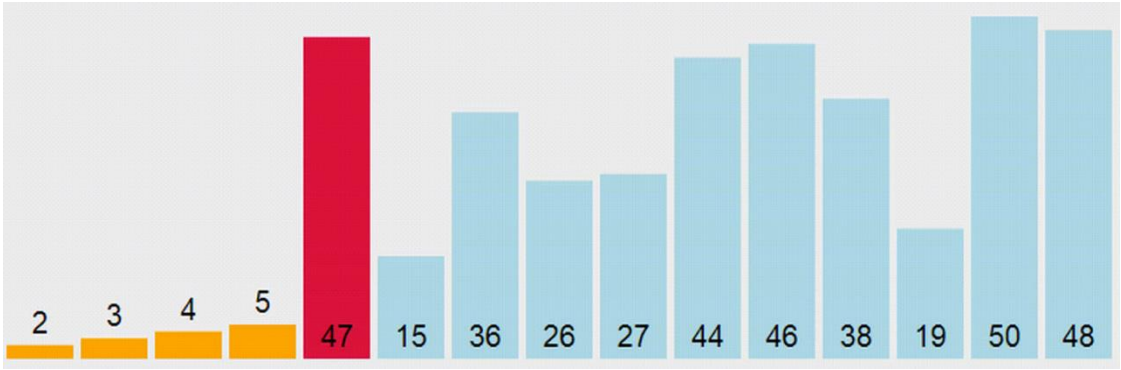


Сортировка поиском (выбором)



Сортировка выбором (поиском)

```
public static int[] selectionSort(int[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        int index = i;
        for (int j = i + 1; j < array.length; j++) {
            if (array[j] < array[index]) {
                index = j;
            }
        }
        int min = array[index];
        array[index] = array[i];
        array[i] = min;
    }
    return array;
}
```



Задачи – часть 1

1. Создать метод нахождения индекса минимального элемента для части массива.
Использовать его в реализации алгоритма сортировки выбором.

2. Подсчитать и вывести на экран количество пар элементов, которые поменяли свои места во время сортировки выбором

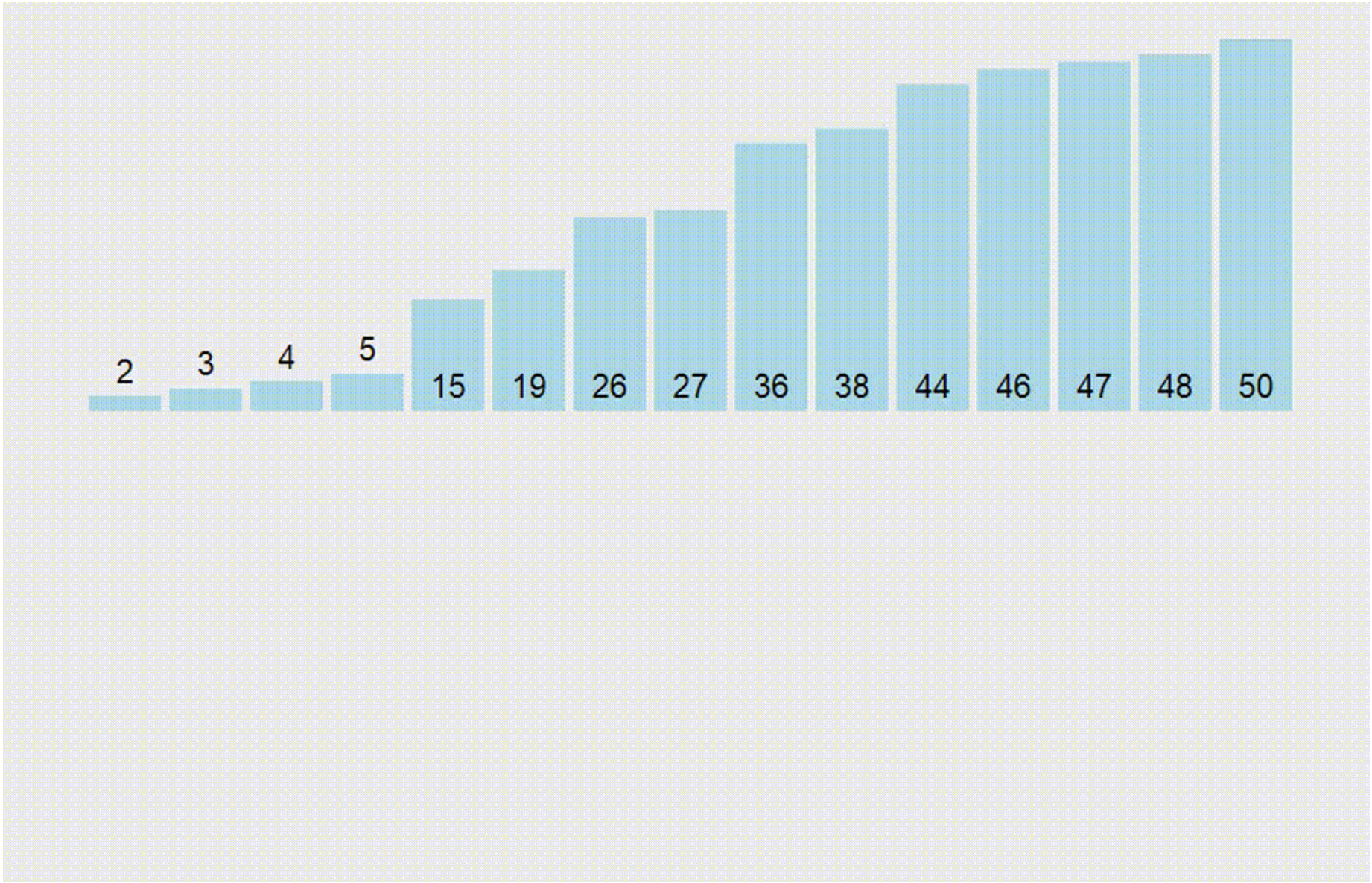
3. Двумерный массив n x m элементов заполнить случайными числами.
Создать и протестировать метод для сортировки выбором одного столбца такого массива.
Отсортировать все столбцы массива и вывести его на экран

4. Одномерный массив заполнить случайными вещественными числами из диапазона [-10; 10].
Отсортировать его элементы с четными индексами (0, 2, 4,…) по возрастанию, а элементы с нечетными индексами (1, 3, 5,…) по убыванию. Использовать сортировку выбором

5. Говорят, что массив отсортирован по max-min значениям, если первый элемент массива является максимальным, второй - минимальным, третий - вторым максимумом и так далее.
Используя принцип сортировки выбором реализуйте сортировку max-min.
Пример:
Входные данные: 1 2 3 4 5
Выходные: 5 1 4 2 3

6. Реализуйте двунаправленный вариант сортировки выбором, который находит минимальное и максимальное значения в массиве за каждый проход и устанавливает их на свои места.
Алгоритм делит массив на три подмассива:
1) отсортированные минимумы;
2) несортированные;
3) отсортированные максимумы.

Сортировка вставками

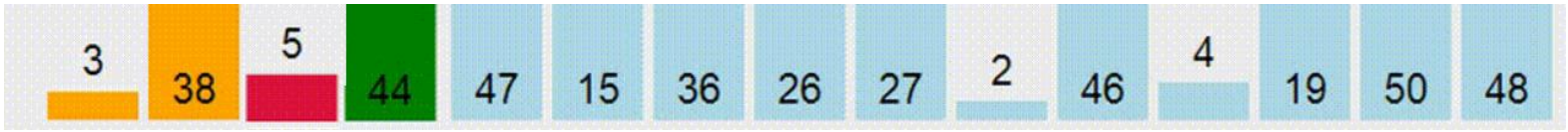


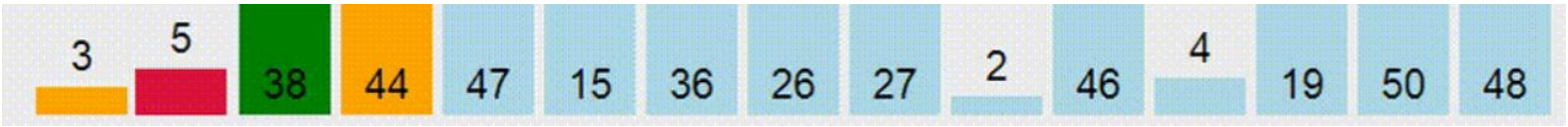
Сортировка вставками

уже упорядочены,
но не все элементы массива рассмотрены

еще не рассматривались

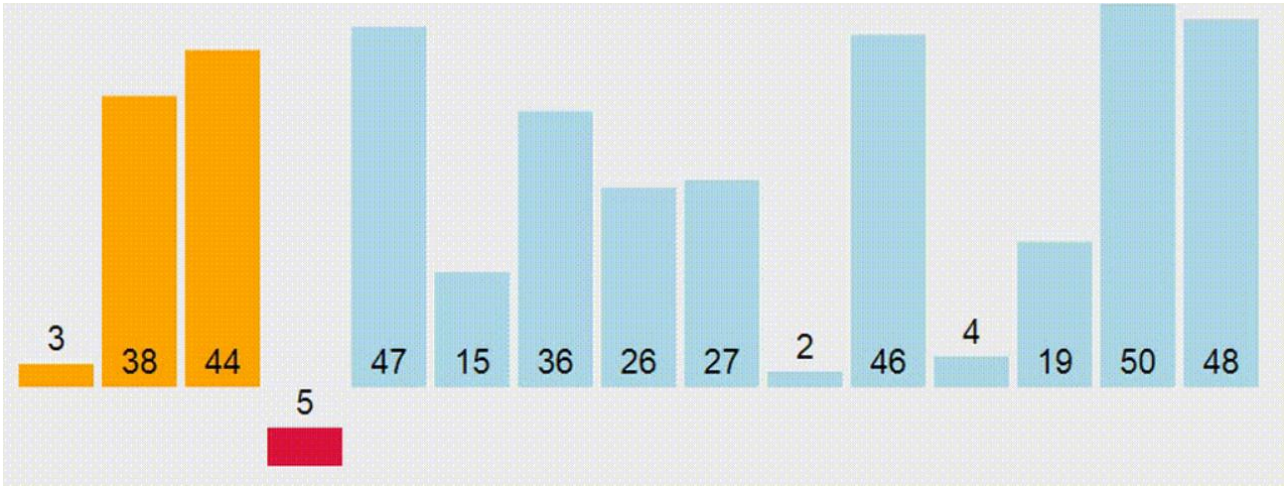
← Двигаем его влево, пока он меньше левого соседа





Сортировка вставками

```
public static int[] insertSort(int[] array) {
    for (int i = 1; i < array.length; i++) {
        int elem = array[i];
        int j = i - 1;
        while (j >= 0 && array[j] > elem) {
            array[j + 1] = array[j];
            j--;
        }
        array[j + 1] = elem;
    }
    return array;
}
```



Задачи – часть 2

7. Двумерный массив n x m элементов заполнить случайными числами.
Создать и протестировать метод для сортировки вставками одного стобца такого массива.
Отсортировать все стобцы массива и вывести его на экран

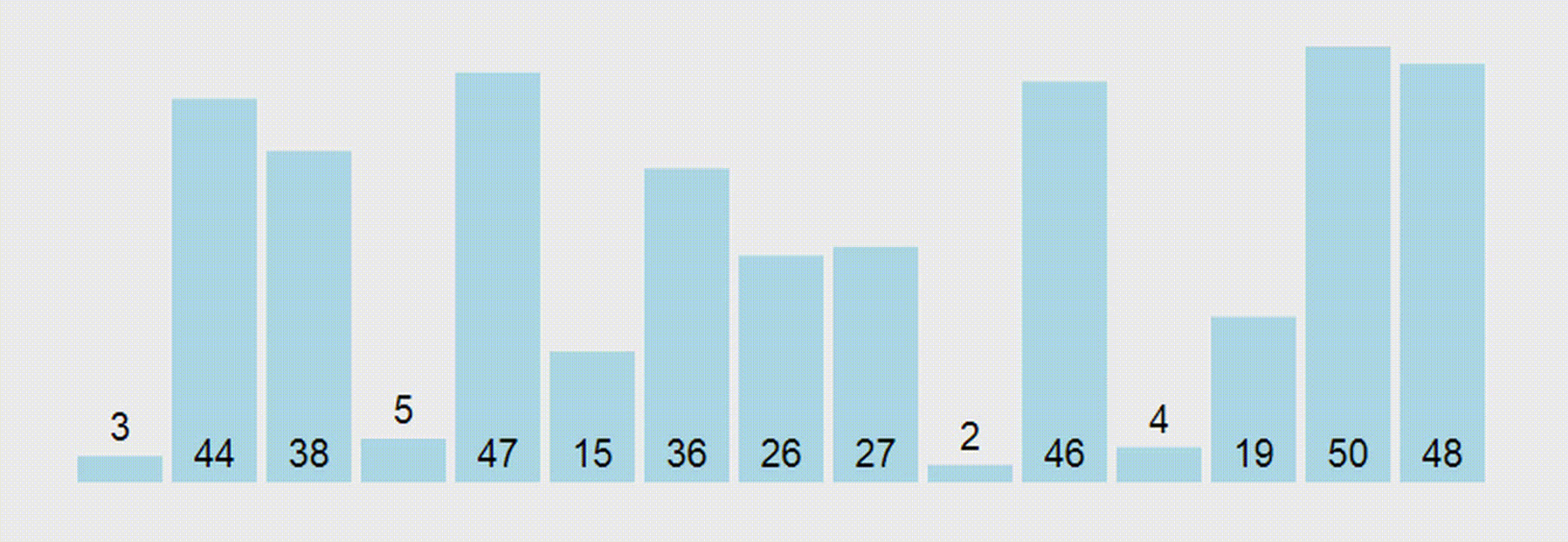
8. Массив целых чисел сортируется вставками в порядке убывания. Выведите количество элементов, сдвинутых со своего изначального места в ходе этой сортировки

Пояснение (см рис.):

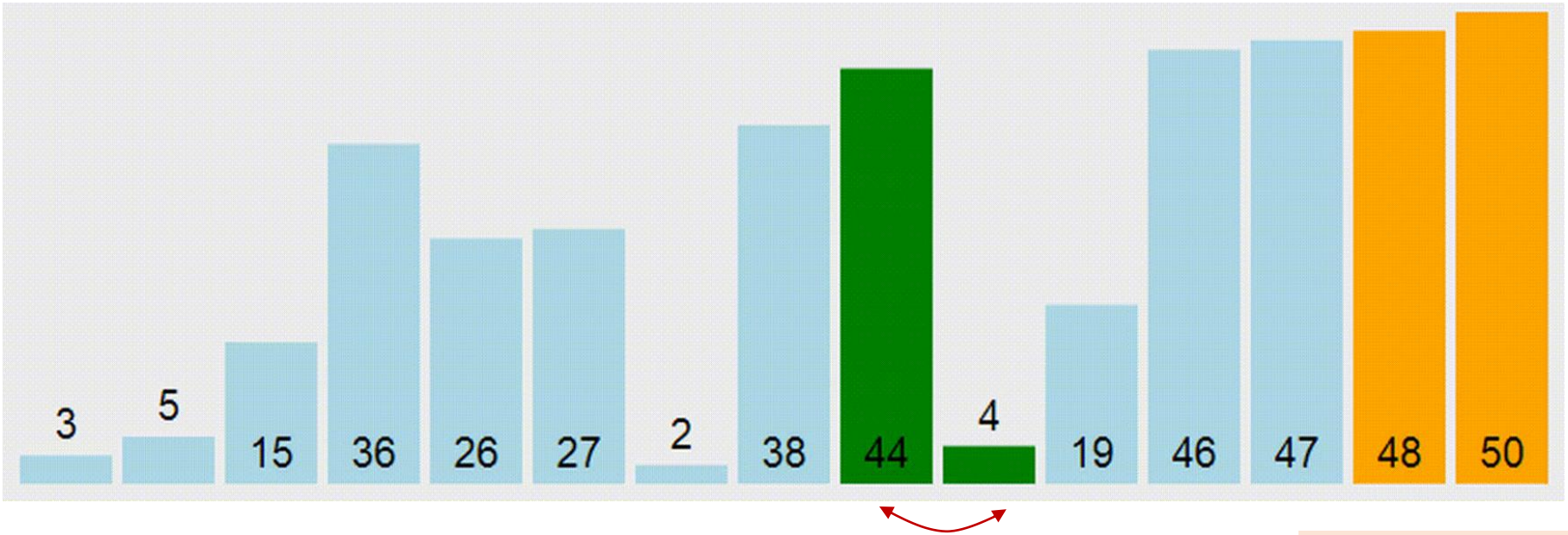
а) число 21 не сдвигается
б) для числа 24 делается перемещение,
значит +1 к счетчику сдвинутых чисел

Пример 1:
ввод: 50 40 30 10 20
вывод: 1
Пример 2:
ввод: 30 40 20 5 10
вывод: 2
Пример 3
ввод: 5 2 9 1 2 4 9 5
вывод: 5

Пузырьковая сортировка



Пузырьковая сортировка

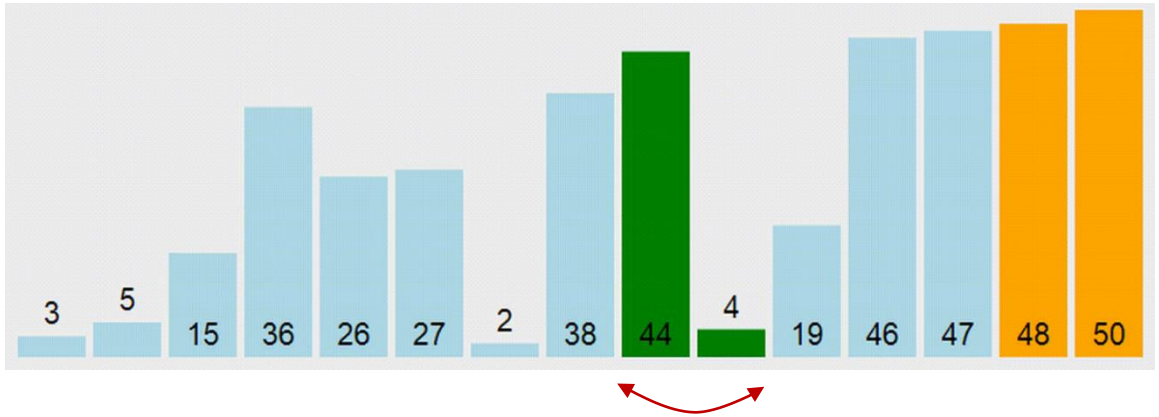


Если пара не упорядочена,
то меняем их местами

Уже «всплыли» самые большие

Пузырьковая сортировка

```
public static int[] bubbleSort (int[] array) {
    boolean f = false;
    int i = 0;
    while ( !f ) {
        f = true;
        for (int j = 0; j < array.length - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                int t = array[j];
                array[j] = array[j + 1];
                array[j + 1] = t;
                f = false;
            }
        }
        i++;
    }
    return array;
}
```



Задачи – часть 3

9. Одномерный массив заполнить случайными вещественными числами. Отсортировать его левую половину по убыванию, а правую половину по возрастанию. Использовать сортировку пузырьком

10. Двумерный массив n x m элементов заполнить случайными числами.
Создать и протестировать метод для сортировки пузырьком одного стобца такого массива.
Отсортировать все стобцы массива и вывести его на экран

11. Числовой массив сортируется пузырьком в порядке убывания. Выведите количество перестановок (обмен двух чисел, swap), которое надо выполнить, чтобы полностью его отсортировать.

Пример 1:
ввод: **5 4 3** 1 2
вывод: 1
Пример 2:
ввод: **8 3 4 6 5 2 1**
вывод: 5