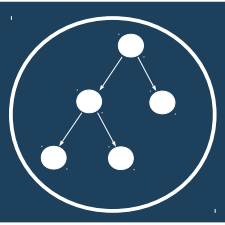


Data Structures and Algorithms

Алгоритмы. Сортировка пузырьком.

Сведение о алгоритме

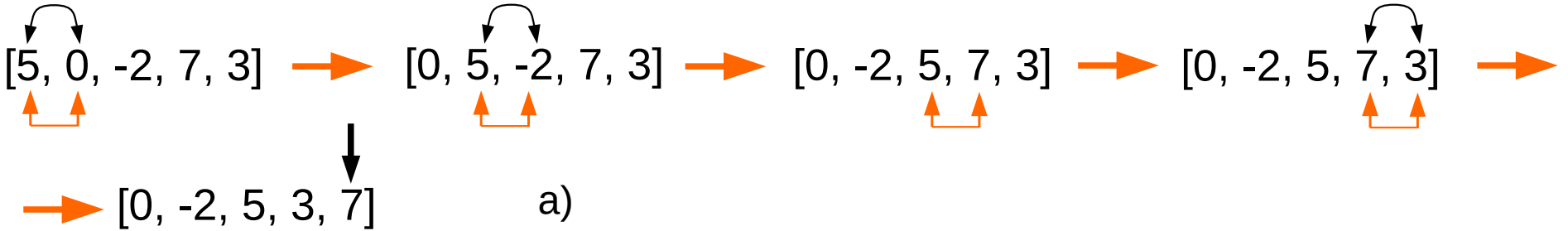
Алгоритм сортировки выбором.

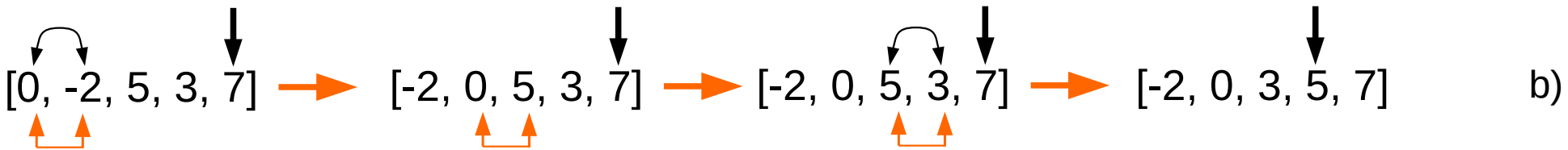
Сложность по времени в наихудшем случае	$O(n^2)$
Затраты памяти	$O(n)$

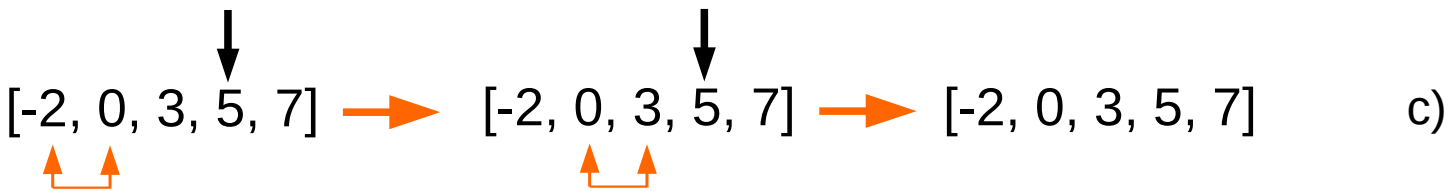
Принцип работы алгоритма

1) Последовательность разбивается на две части. Отсортированную и не отсортированную. В качестве отсортированной части обычно выбирается правая часть последовательности.
2) Выполняется проход по не отсортированной части. И выполняется попарное сравнение элементов последовательности. Если первый элемент пары больше второго элемента, то происходит их обмен. В результате такого прохода максимальный элемент не отсортированной части последовательности попадает в ее конец. Он становиться первым членом отсортированной части последовательности.
3) Алгоритм прекращает свою работу в случае отсутствия обменов при проходе по не отсортированной части последовательности.

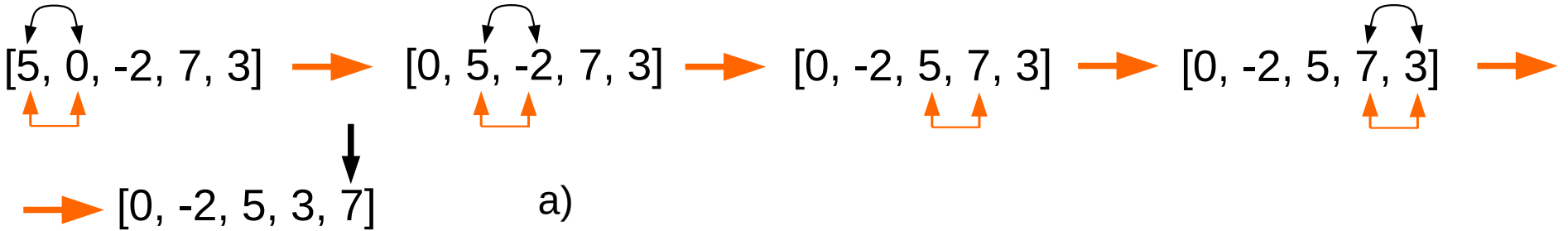
Графическая иллюстрация работы алгоритма





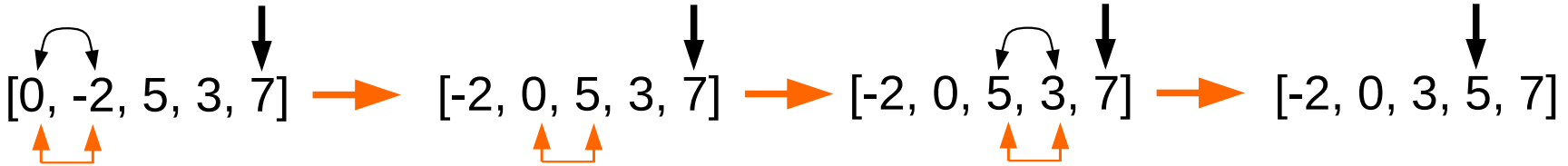


Графическая иллюстрация работы алгоритма



Вначале вся последовательность является не отсортированной. Выполняется попарный проход по ней (оранжевые стрелки внизу). Если первый элемент пары, больше второго то выполняется их обмен (черные стрелки вверху). Как можно видеть всего за один проход было выполнено 3 обмена. Максимальный элемент оказывается последним элементом не сортированной части. Он же становится первым элементом отсортированной части (указатель на начало отсортированной части изображен вертикальной стрелкой).

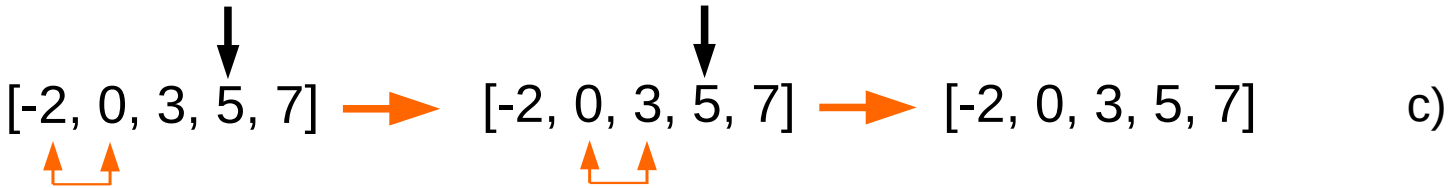
Графическая иллюстрация работы алгоритма



b)

При следующем проходе. Действия повторяются. Только теперь размер не отсортированной части последовательности меньше. При этом проходе было выполнено два обмена. После этого последний элемент не отсортированной части становится первым элементом отсортированной части.

Графическая иллюстрация работы алгоритма



Выполняется попарный проход по не отсортированной части последовательности. За весь проход не было выполнено ни одного обмена. Алгоритм завершается. Как можно видеть последовательность отсортирована.

Python

Реализация алгоритма на Python

Реализация алгоритма на Python

```
list_1 = [5, 0, -2, 7, 3]

sorted_index = len(list_1)

while True:
    number_of_swap = 0
    for i in range(0, sorted_index-1):
        if list_1[i] > list_1[i+1]:
            list_1[i], list_1[i+1] = list_1[i+1], list_1[i]
            number_of_swap += 1
    sorted_index -= 1
    if number_of_swap == 0:
        break

print(list_1)
```

Указатель на отсортированную часть

Счетчик обменов

Перебор индексов не отсортированной части

Обмен

Увеличение количества обменов

Если обменов не было заканчиваем

Java

Реализация алгоритма на Java

Реализация алгоритма на Java

```
int[] array = new int[] { 5, 0, -2, 7, 3 };

int sortedIndex = array.length;

int numberOfSwap = 1;

while (numberOfSwap > 0) {
    numberOfSwap = 0;
    for (int i = 0; i < sortedIndex - 1; i++) {
        if (array[i] > array[i + 1]) {
            int temp = array[i];
            array[i] = array[i + 1];
            array[i + 1] = temp;
            numberOfSwap += 1;
        }
    }
    sortedIndex -= 1;
}

System.out.println(Arrays.toString(array));
```

Указатель на отсортированную часть

Счетчик обменов

Если обменов не было заканчиваем

Перебор индексов не отсортированной части

Обмен

Увеличение количества обменов

Сдвиг указателя на отсортированную часть

Список литературы

1) Ананий Левитин. Алгоритмы: введение в разработку и анализ. : Пер. с англ. — М. : Издательский дом "Вильяме", 2006. — 576 с. : ил. — Парал. тит. Англ. ISBN 5-8459-0987-2. Стр. [144-146]
2) Стивенсон Род. Алгоритмы. Теория и практическое применение — М: Издательство «Э», 2016 — 544. ISBN 978-5-699-81729-0. Стр. [139-142]