

Data Structures and Algorithms

Алгоритмы. Сортировка вставками.



Сведение о алгоритме

Алгоритм сортировки вставками.

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

Затраты памяти $O(n)$

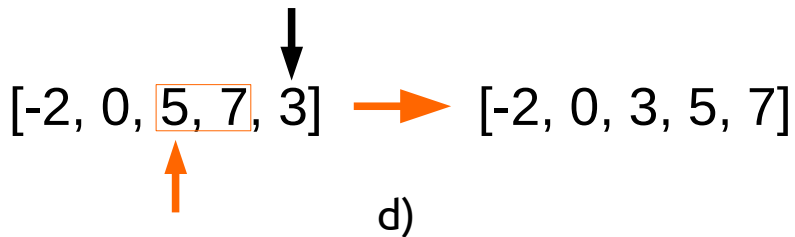
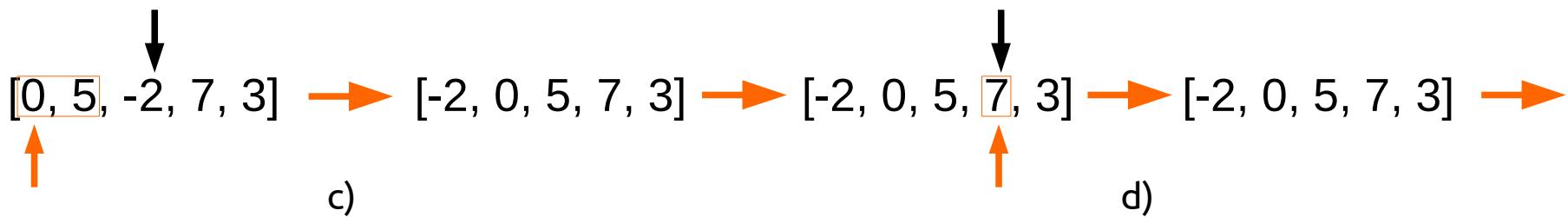
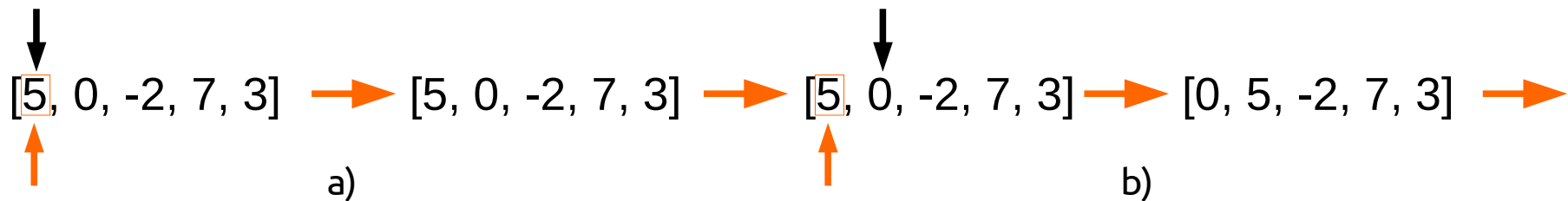


Принцип работы алгоритма

- 1) Последовательность разбивается на две части. Отсортированную и не отсортированную. В качестве отсортированной части обычно выбирается левая часть последовательности.
- 2) Выбирается первый элемент не отсортированной последовательности и для него находится позиция в отсортированной части последовательности. Правило нахождения позиции для элемента со значением X :
$$a_n \leq X \leq a_{n+1}$$
- 3) Значение устанавливается на найденную позицию со сдвигом элементов стоящим справа от найденной позиции.
- 4) Алгоритм продолжается до исчерпания не отсортированной части последовательности.

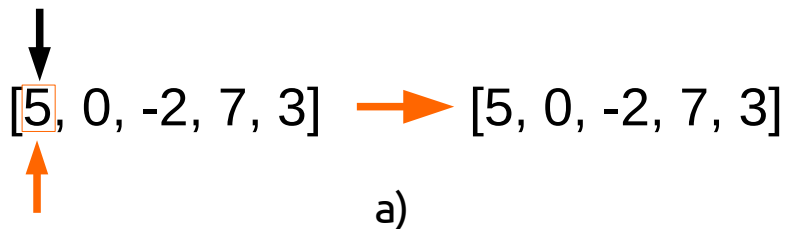


Графическая иллюстрация работы алгоритма





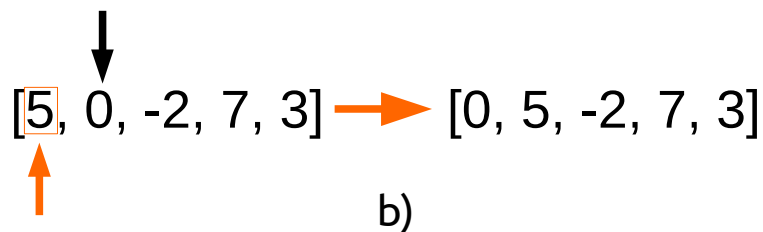
Графическая иллюстрация работы алгоритма



Вначале вся последовательность является не отсортированной. Поэтому первый ее элемент сразу становится первым членом отсортированной части. Довольно часто этот шаг просто опускают начиная со второго члена последовательности.



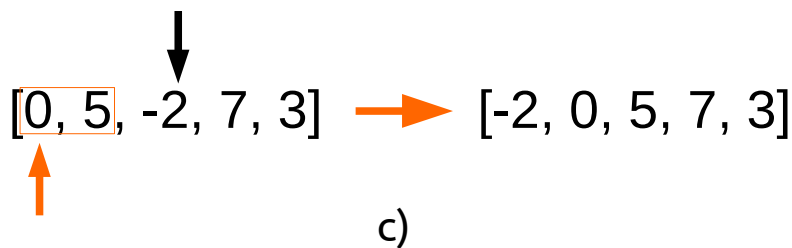
Графическая иллюстрация работы алгоритма



Выбираем первый член не отсортированной последовательности (черная стрелка) и находим для него позицию (оранжевая стрелка). Ставим элемент на эту позицию. Сдвигаем часть отсортированной позиции (выделена оранжевым прямоугольником) на один шаг вправо.



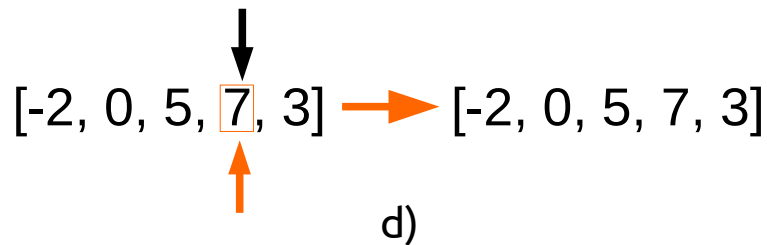
Графическая иллюстрация работы алгоритма



Выбираем первый член не отсортированной последовательности (черная стрелка) и находим для него позицию (оранжевая стрелка). Ставим элемент на эту позицию. Сдвигаем часть отсортированной позиции (выделена оранжевым прямоугольником) на один шаг вправо.



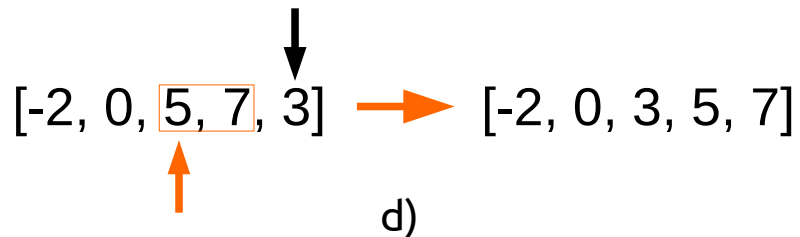
Графическая иллюстрация работы алгоритма



Выбираем первый член не отсортированной последовательности (черная стрелка) и находим для него позицию (оранжевая стрелка). Так как позиция совпадает с положением элемента то ничего не делаем.



Графическая иллюстрация работы алгоритма



Выбираем первый член не отсортированной последовательности (черная стрелка) и находим для него позицию (оранжевая стрелка). Ставим элемент на эту позицию. Сдвигаем часть отсортированной позиции (выделена оранжевым прямоугольником) на один шаг вправо. Не отсортированная часть исчерпана, алгоритм закончен.



Реализация алгоритма на Python



Реализация алгоритма на Python

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

```
for i in range(1, len(list_1)):
    paste_element = list_1[i]
    while i > 0 and list_1[i-1] > paste_element:
        list_1[i] = list_1[i-1]
        i = i-1
    list_1[i] = paste_element
```

Указатель на первый элемент не отсортированной части

Поиск позиции для вставки

Ставим элемент на нужную позицию

```
print(list_1)
```



Java

Реализация алгоритма на Java



Реализация алгоритма на Java

```
int[] array = new int[] { 5, 0, -2, 7, 3 };  
  
for (int i = 0; i < array.length; i++) {  
    int pasteElement = array[i];  
    int j;  
  
    for (j = i; j > 0; j--) {  
        if (array[j - 1] <= pasteElement) {  
            break;  
        }  
        array[j] = array[j - 1];  
    }  
    array[j] = pasteElement;  
}  
  
System.out.println(Arrays.toString(array));
```

Указатель на первый элемент не отсортированной части

Берем первый элемент не отсортированной части

Поиск позиции для вставки

Сдвиг отсортированной последовательности вправо

Ставим элемент на нужную позицию



Список литературы

- 1) Ананий Левитин. Алгоритмы: введение в разработку и анализ. : Пер. с англ. — М. : Издательский дом "Вильямс", 2006. — 576 с. : ил. — Парал. тит. Англ. ISBN 5-8459-0987-2. Стр. [206-209]
- 2) Стивенсон Род. Алгоритмы. Теория и практическое применение — М: Издательство «Э», 2016 — 544. ISBN 978-5-699-81729-0. Стр. [136-138]