

## Сортировка вставками

Рассмотрим реализацию алгоритма сортировки одномерного массива методом вставок.

*⟨insertionSort.py⟩*≡

```
def insertion_sort(arr):  
    ⟨Основной цикл⟩
```

Создаём цикл, проходящий по массиву `arr`, начиная со второго элемента, так как считается, что первый элемент уже отсортирован

*⟨Основной цикл⟩*≡

```
for i in range(1, len(arr)):  
    ⟨Сохранение текущего элемента⟩  
    ⟨Сохранение индекса предыдущего числа⟩  
    ⟨Поиск позиции для вставки⟩  
    ⟨Вставка элемента⟩
```

Сохраним число из массива с индексом `i` в переменную `item`.

*⟨Сохранение текущего элемента⟩*≡

```
item = arr[i]
```

Сохраним индекса предыдущего числа в переменную `j`

*⟨Сохранение индекса предыдущего числа⟩*≡

```
j = i - 1
```

С помощью цикла `while` находим такую позицию в массиве, при которой число в переменной `item` будет меньше чем `arr[j]`.

*⟨Поиск позиции для вставки⟩*≡

```
while j >= 0 and arr[j] > item:  
    ⟨Сдвиг элементов в массиве⟩  
    j -= 1
```

При каждой итерации цикла `while` сдвигаем элемент массива с индексом `j` на одну позицию вперёд.

*⟨Сдвиг элементов в массиве⟩*≡

```
arr[j + 1] = arr[j]
```

Производим вставку элемента в массив по индексу `j + 1`

*⟨Вставка элемента⟩*≡

```
arr[j + 1] = item
```

1

---

<sup>1</sup>Разработал Ворожейкин Д.А. ИКБО-01-19