Bubble Sort
Insertion Sort
Selection Sort

Bubble Sort

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

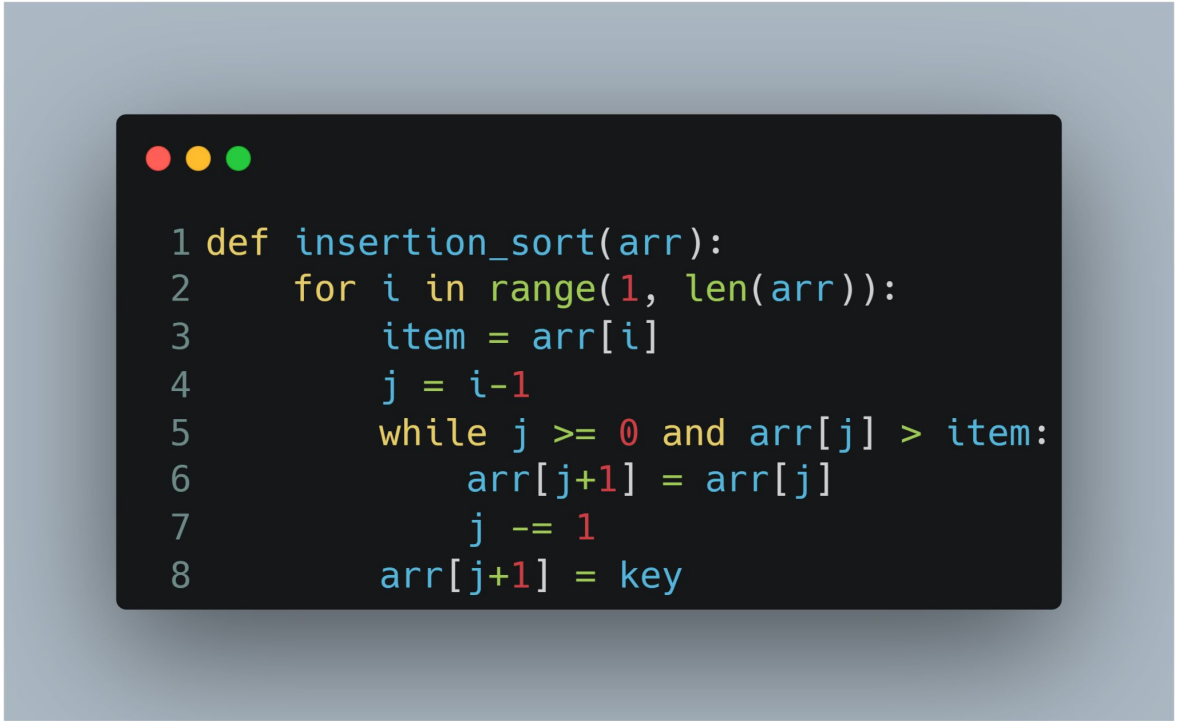
Best Case : Already Sorted. Time Complexity : O(n).
Average Case : Time Complexity : $O(n^2)$.
Worst Case : Sorted in Reverse Order. Time Complexity : $O(n^2)$.

Bubble Sort

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        check = False
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                check = True
        if check == False:
            break
```

Best Case : Already Sorted. Time Complexity : O(n).
Average Case : Time Complexity : $O(n^2)$.
Worst Case : Sorted in Reverse Order. Time Complexity : $O(n^2)$.

Insertion Sort



```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        item = arr[i]
        j = i-1
        while j >= 0 and arr[j] > item:
            arr[j+1] = arr[j]
            j -= 1
        arr[j+1] = key
```

Best Case : Already Sorted. Time Complexity : O(n).
Average Case : Time Complexity : $O(n^2)$.
Worst Case : Sorted in Reverse Order. Time Complexity : $O(n^2)$.

Selection Sort

```python
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_index = i
        for j in range(i+1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        arr[i], arr[min_index] = arr[min_index], arr[i]
```

Time Complexity : $O(n^2)$ in all cases.