CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

Insertion Sort

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

What will we Cover?

- Insertion sort algorithm
- Understanding the efficiency of the algorithm using big-O notation

CITY
UNIVERSITY OF LONDON
EST 1894

Insertion Sort

- Insertion sort algorithm splits the list into sorted part and unsorted part.
- Elements form the unsorted part are inserted into the correct positions in the sorted part.

Tracing the Behaviour of the Insertion Sort

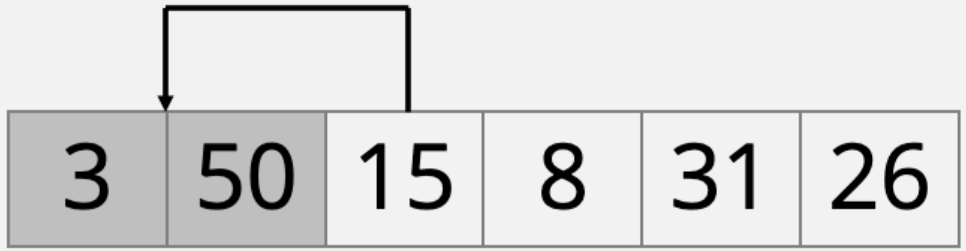
Initially, the sorted sub-list contains the first element in the list.

50	3	15	8	31	26

Tracing the Behaviour of the Insertion Sort

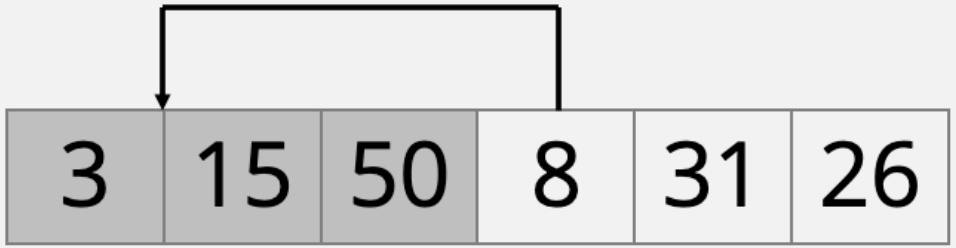
The sorted sub-list is
[50] Insert 3 into the list

Tracing the Behaviour of the Insertion Sort



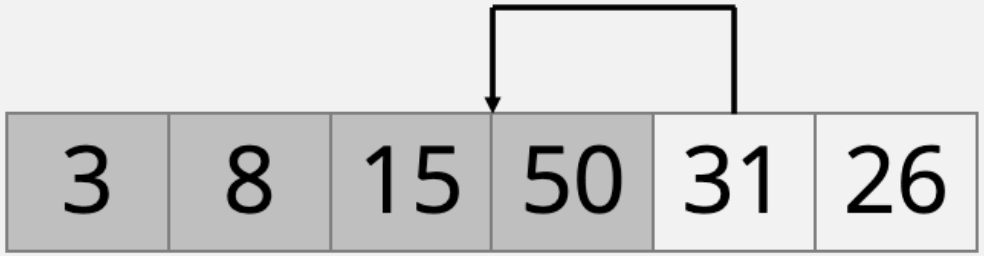
The sorted sub-list is [3, 50] Insert 15 into the list

Tracing the Behaviour of the Insertion Sort



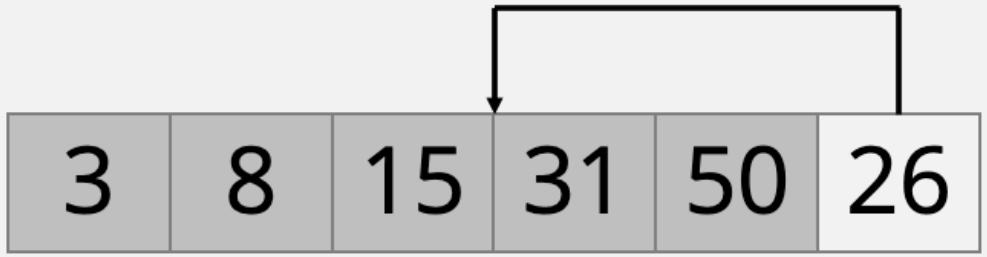
The sorted sub-list is [3, 15, 50] Insert 8 into the list

Tracing the Behaviour of the Insertion Sort



The sorted sub-list is [3, 8, 15, 50] Insert 31 into the list

Tracing the Behaviour of the Insertion Sort



The sorted sub-list is [3, 8, 15, 31, 50] Insert 26 into the list

Tracing the Behaviour of the Insertion Sort

The entire list is now sorted

3	8	15	26	31	50

Insertion Sort Algorithm

```
1.  n = len(list)
2.  for i in range(n-1)
3.      j = i
4.      while j > 0 and list[j-1] > list[j] do
5.          temp = list[j]
6.          list[j] = list[j-1]
7.          list[j-1] = temp
8.          j = j - 1
```

CITY
UNIVERSITY OF LONDON
EST 1894

Insertion Sort Example

```
def insertion_sort(list1):
    # this function sorts the list using insertion sort algorithm
    n = len(list1)
    for i in range(n):
        j = i
        while j > 0 and list1[j-1] > list1[j]:
            temp = list1[j]
            list1[j] = list1[j-1]
            list1[j-1] = temp
            j = j-1

input_string = input("Enter your numbers, then press enter: ")
split_input = input_string.split()
numbers = [int(n) for n in split_input]

insertion_sort(numbers)
print("sorted list:", numbers)
```

```
Enter your numbers, then press enter: 9 4 6 1 5 0 3 7 2 8
sorted list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Analysis

- Sorts elements one by one
- Works best on a partially sorted list
- Easy to implement and works best for keeping a list sorted while inserting new elements
- Insertion sort uses a nested loop, so in the worst-case scenario, when we have to execute the inner loop **n** times, we perform **$O(n^2)$** comparisons
- The big-O notation for insertion sort is **$O(n^2)$**

Try It Yourself

Write a program in python environment that takes a string as an input and sorts in alphabetical order using the insertion sort algorithm above