**Bubble Sort**

Bubble Sort is the simplest [sorting algorithm](https://www.geeksforgeeks.org/sorting-algorithms/) that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

**Bubble Sort Algorithm:**

*In this algorithm,*

* *traverse from left and compare adjacent elements and the higher one is placed at right side.*
* *In this way, the largest element is moved to the rightmost end at first.*
* *This process is then continued to find the second largest and place it and so on until the data is sorted.*

**Advantages of Bubble Sort:**

* Bubble sort is easy to understand and implement.
* It does not require any additional memory space.
* It is a stable sorting algorithm, meaning that elements with the same key value maintain their relative order in the sorted output.

**Disadvantages of Bubble Sort:**

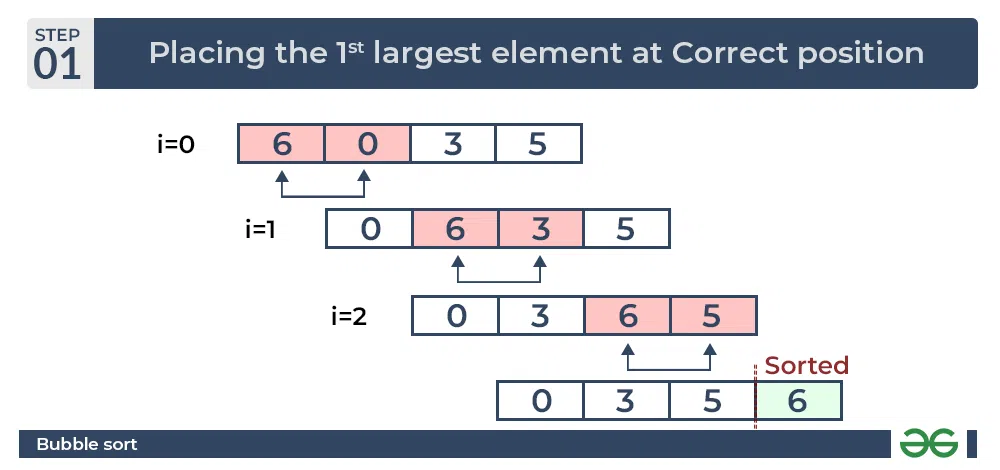
* Bubble sort has a time complexity of O(N2) which makes it very slow for large data sets.
* Bubble sort is a comparison-based sorting algorithm, which means that it requires a comparison operator to determine the relative order of elements in the input data set. It can limit the efficiency of the algorithm in certain cases.

**How does Bubble Sort Work?**

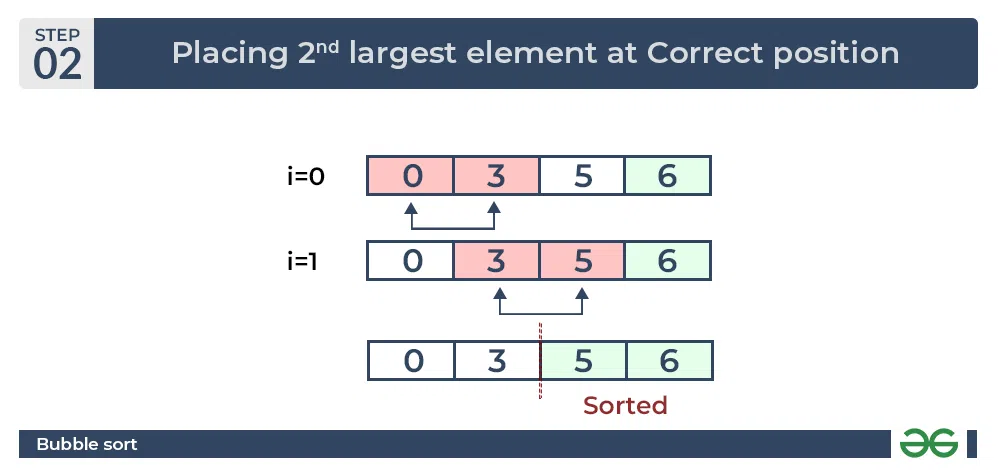
Let us understand the working of bubble sort with the help of the following illustration:

*Input: arr[] = {6, 3, 0, 5}*

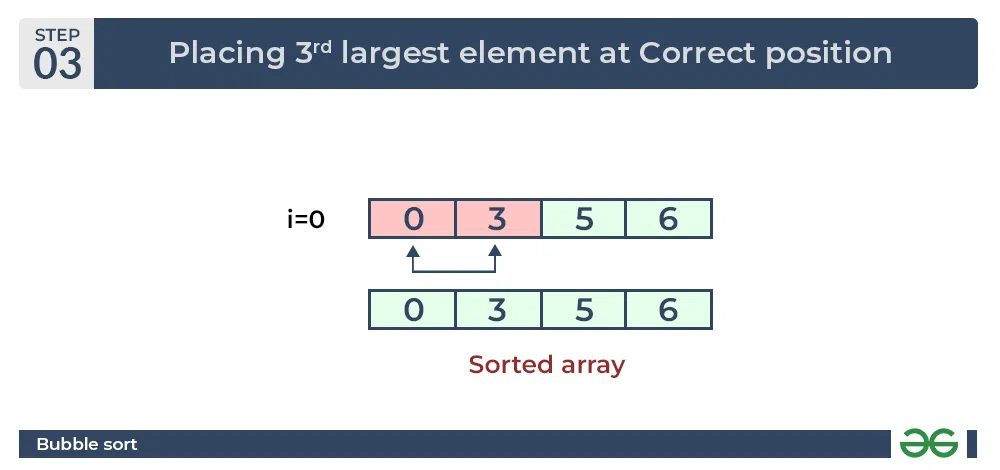
***First Pass:***

*The largest element is placed in its correct position, i.e., the end of the array. *

***Second Pass:***

*Place the second largest element at correct position.*

***Third Pass:***

*Place the remaining two elements at their correct positions****.***

**Done by: Anas Abdullah.**

**Teacher: Ziad Mansor.**

**Source Code**

**# Optimized Python program for implementation of Bubble Sort**

def bubbleSort(arr):

n = len(arr)

**# Traverse through all array elements**

for i in range(n):

swapped = False

**# Last i elements are already in place**

for j in range(0, n-i-1):

**# Traverse the array from 0 to n-i-1**

**# Swap if the element found is greater**

**# than the next element**

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

swapped = True

if (swapped == False):

break

**# Driver code to test above**

if \_\_name\_\_ == "\_\_main\_\_":

arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print("Sorted array:")

for i in range(len(arr)):

print("%d" % arr[i], end=" ")

**Output**

Sorted array:

11 12 22 25 34 64 90