



# Understanding Sorting

Based on CBSE Curriculum  
Class -11

## Chapter-10

By-

Neha Tyagi

PGT CS

KV 5 Jaipur II Shift

Jaipur Region

# INTRODUCTION

- Generally, sorting means to categorize the objects as per their groups.
- A Computer system needs to work with different types of elements and sorting means to arrange these elements in an order.
- Order of sorting can be ascending or descending.
- In this chapter we will study sorting in details and will see two sorting techniques-
  1. Bubble Sort
  2. Insertion Sort

# What is Sorting

- “In computer, sorting means to arrange elements in an order, order can be ascending or descending”.
- Consider the following sequence -  
Seq = [6,8,9,1,3,4]  
Its new sequence arranged in ascending order will be-  
Seq = [1,3,4,6,8,9 ]  
And its new sequence arranged in descending order will be-  
Seq = [9,8,6,4,3,1]
- There are various techniques of sorting like- *Selection Sort, Bubble Sort, Insertion Sort, Heap Sort, Quick Sort* etc.
- Only two sorting techniques are in our syllabus-
- **Bubble sort**
- **Insertion sort**

# Bubble Sort

- In Bubble Sort, we compare two adjacent elements and if the elements are not in correct order then these elements got exchanged. After every pass the biggest element secures its correct position.

```
lst=[10,4,6,18,2,17,3]
l=len(lst)
for i in range(l):
    for j in range(0,l-i-1):
        if lst[j]>lst[j+1]:
            temp=lst[j]
            lst[j]=lst[j+1]
            lst[j+1]=temp
print("After sorting the list is : ")
print(lst)
```

The concept of this program can be easily understood by dry run and Python debugger tool can also be used for this.

OUTPUT:

```
After sorting the list is :
[2, 3, 4, 6, 10, 17, 18]
```

# Detection of number of operations in Bubble Sort

- Number of Operations is an important issue because more number of operations means more CPU time will be required.
- Two programs with different logic can produce the same output with different number of operations.
- Consider the following code which has assignment, comparison, swapping in it.


```
1. lst=[10,4,6,18,2,17,3]
2. print("Original List is : ",lst)
3. l=len(lst)
4. for i in range(l):
5.     for j in range(0,l-i-1):
6.         if lst[j]>lst[j+1]:
7.             temp=lst[j]
8.             lst[j]=lst[j+1]
9.             lst[j+1]=temp
10. print("After sorting the list is : ")
11. print(lst)
```

Python provides another code also for this (Swap) -  
**lst [ j ], lst [ j+1 ] = lst [ j + 1 ], lst [ j ]**

# Detection of number of operations in Bubble Sort

To get the number of operations-

- In line 1, line 2, line 3, Operations = 3
- Number of operations from line 4 to 9 = ?
- In line 10, operation =1
- Now, need to detect number of operations in outer for loop
- Outer loop will run 7 times for 7 values given in example-
  - In line 4 , operation=1
  - Inner for loop will run 6 times (n-i-1) for line 5 to 9.
  - It will run 6 times for the first time, 5 times for the second time, 4 times for the third time and so on.
  - In line 5, operation=1.
  - In line 6,7,8,9, operations=4

- Therefore, total operation for outer for loop  total 5
- In 1<sup>st</sup> iteration (i=0) = 1 op (line 4) + 5 ops (Line 5-9) X 6 times = 31 ops
- In 2<sup>nd</sup> iteration (i=1) = 1 op (line 4) + 5 ops (Line 5-9) X 5 times = 26 ops
- In 3<sup>rd</sup> iteration (i=2) = 1 op (line 4) + 5 ops (Line 5-9) X 4 times = 21 ops
- In 4<sup>th</sup> iteration (i=3) = 1 op (line 4) + 5 ops (Line 5-9) X 3 times = 16 ops
- In 5<sup>th</sup> iteration (i=4) = 1 op (line 4) + 5 ops (Line 5-9) X 2 times = 11 ops
- In 6<sup>th</sup> iteration (i=5) = 1 op (line 4) + 5 ops (Line 5-9) X 1 times = 6 ops
- In 7<sup>th</sup> iteration (i=6) = 1 op (line 4) + 5 ops (Line 5-9) X 0 times = 1 ops
- Total 112 operations will be for outer loop.

**In total, number of operations 3+112 +1 = 116**

# Detection of number of operations in Bubble Sort

- If we consider following program-

```
1. lst=[10,4,6,18,2,17,3]
2. print("Original List is : ",lst)
3. l=len(lst)
4. for i in range(l):
5.     for j in range(0,l-i-1):
6.         if lst[j]>lst[j+1]:
7.             lst [ j ], lst [ j + 1 ] = lst [ j + 1 ], lst [ j ]
8. print("After sorting the list is : ",lst)
```

- For 7 values of N, number of operations=74.
- For 8 values of N, number of operations=96, For 9 values of N, number of operations=121 For 6 values of N, number of operations=55.
- Number of comparisons for total values of N will be  $(N-1)+(N-2)+(N-3) + \dots + (2)+(1)$  means  $N^2$ .
- Number of swappings for total values of N will be  $(N-1)+(N-2)+(N-3) + \dots + (2)+(1)$  means  $N^2$ .
- In a Best case, when sequence got fully sorted, number of operations will be  $N^2$ .
- In a Worst case, when every element in reverse order, number of operations will be

$$N^2 \text{ (comparison)} + N^2 \text{ (Swappings)} = 2 N^2$$

# Insertion Sort

- In insertion Sort, firstly two first values gets in order.
- We call it as sub-sequence which will be sorted.
- From rest of the unsorted sequence, values will be taken one by one and will be inserted at right position in sorted sub-sequence.
- This process will be continued until we get the ordered sequence. |

```
lst=[15,14,19,2,6,4,5,11,12,10,8,7]
print("Original List is : ", lst)
for i in range(1,len(lst)):
    temp=lst[i]
    j=i-1
    while j>=0 and temp<lst[j]:
        lst[j+1]=lst[j]
        j=j-1
    else:
        lst[j+1]=temp
print("List after sort is : ",lst)
```

OUTPUT:

Original List is : [15, 14, 19, 2, 6, 4, 5, 11, 12, 10, 8, 7]  
List after sort is : [2, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15, 19]

The concept of this program can be easily understood by dry run and Python debugger tool can also be used for this.



# Usage of Bubble Sort and Insertion Sort

- Bubble sort is considered as a very simple and easy sorting algorithm but it is not considered as a very efficient algorithm.
- For a small database, insertion sort is better than bubble sort.
- For the sorting of a file stored in a tape, bubble sort is better because in a tape, data is stored in sequential format.
- Insertion sort is usually used to sort small databases. When the database is almost sorted then this technique work faster as not much memory is needed.
- In small sequences, less number of swapping is there in insertion sort as compared to bubble sort.

# Question-Answers

- What is Sorting? Name some of the sorting techniques.
- What is the concept of Bubble sort?
- What are the basis to select an algorithm?
- What is the concept of Insertion sort?
- Compare insertion sort and bubble sort on the basis of number of Operations.
- In which case insertion sort is also not efficient?
- What are the major dissimilarities between Insertion sort and bubble sort?

# Thank you

please follow us on our blog -

[www.pythontrends.wordpress.com](http://www.pythontrends.wordpress.com)

## एक शुरुआत pythontrends

पाइथन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

लेख/Articles

छायाचित्र/Images

विडियो/Video

अध्यायवार पाठ्यसामग्री/Lesson wise  
Study Material

उपयोगी लिंक्स / Useful Links

पाइथन प्रोग्राम/Python Programs

## नमस्ते दोस्तों ! /Hello Friends!



यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रेजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेज़ी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेज़ी के कठिन शब्दों को

