CBS Department

Sorting Algorithms – Insertion Sort

Soma Soleiman Zadeh

Data Structures & Algorithms (CBS 216)

Spring 2025 - 2026

Week 12

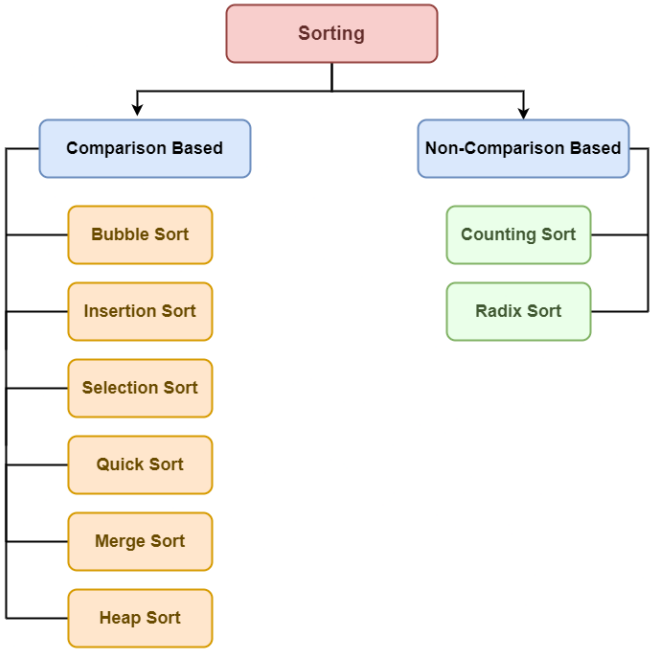
April 22-23, 2026

Outline

- **Insertion Sorting** Algorithm
- When to Use **Insertion Sort**?
- Time Complexity and Space Complexity

Which Sorting Algorithm is the Best?

- Time Complexity
- Space Complexity



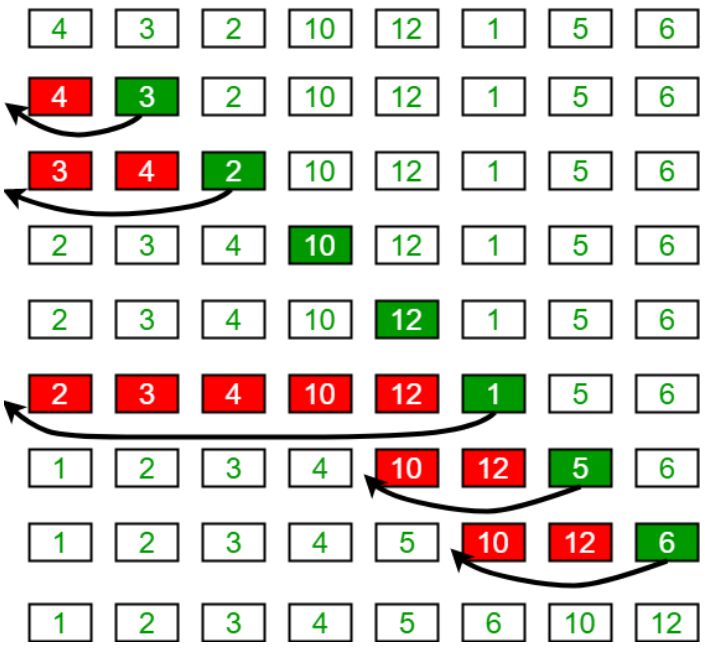
Time Complexity & Space Complexity

Sorting Algorithm	Time Complexity			Space Complexity
	Best case	Average case	Worst case	Worst case
Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Bubble	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Quick	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n)$
Merge	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Heap	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$

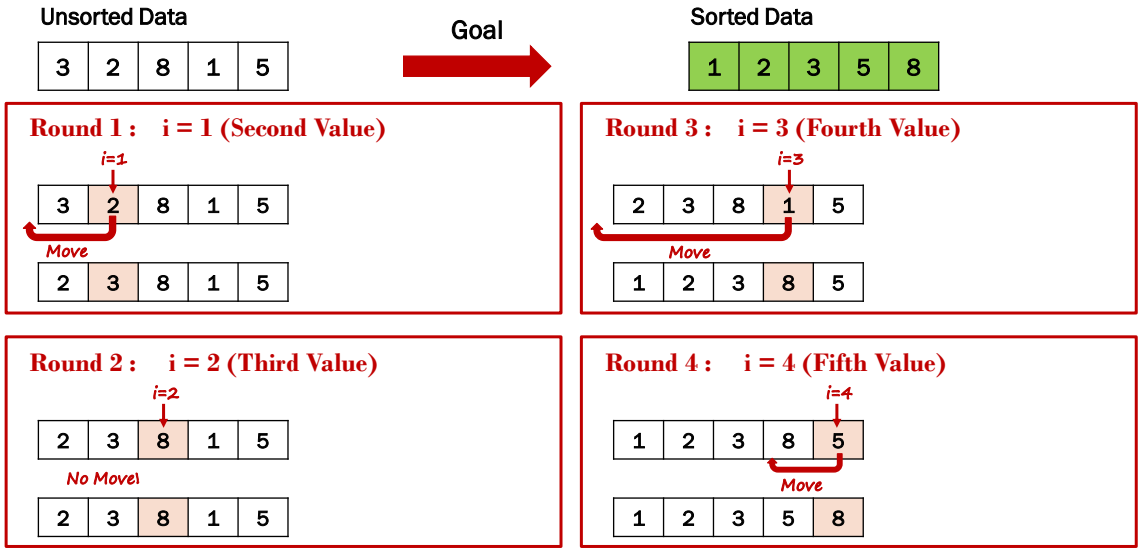
Insertion Sort

- An **insertion sort** compares values in turn, starting with the second value in the list.
  - If this value is greater than the value to the left of it, no changes are made.
  - Otherwise, **this value is repeatedly moved left until it meets a value that is less than it.**
- The sort process then starts again with the next value. This continues until the end of the list is reached.

Insertion Sort



How Insertion Sort Works?



When to Use Insertion Sort?

- **Insertion Sort** is an efficient sorting algorithm for:
  - **Small Datasets**
  - **Nearly Sorted Datasets**
- **Insertion sort** generally performs better than **bubble sort**, in average scenarios.

Insertion Sort Implementation

```
######## Insertion Sort Function #########

def insertionSort(data) :
    n = len(data)
    for i in range( 1, n) :
        temp = data[i]
        j = i
        while j > 0 and data[j-1] >= temp :
            data[j] = data[j-1]
            j -= 1
        data[j] = temp
```

Insertion Sort: Time and Space Complexity

Sorting Algorithm	Time Complexity (Worst Case)	Space Complexity	Stability
Insertion Sort	$O(n^2)$	$O(1)$	**Stable**

- **Simple Sorting Algorithm**
- **Good for Nearly Sorted Data**
- **Slow for Large Datasets**