



# Sorting Algorithms, Bubble Sort

Soma Soleiman Zadeh

Data Structures & Algorithms (CBS 216)

Spring 2025 - 2026

Week 10

April 9, 2026

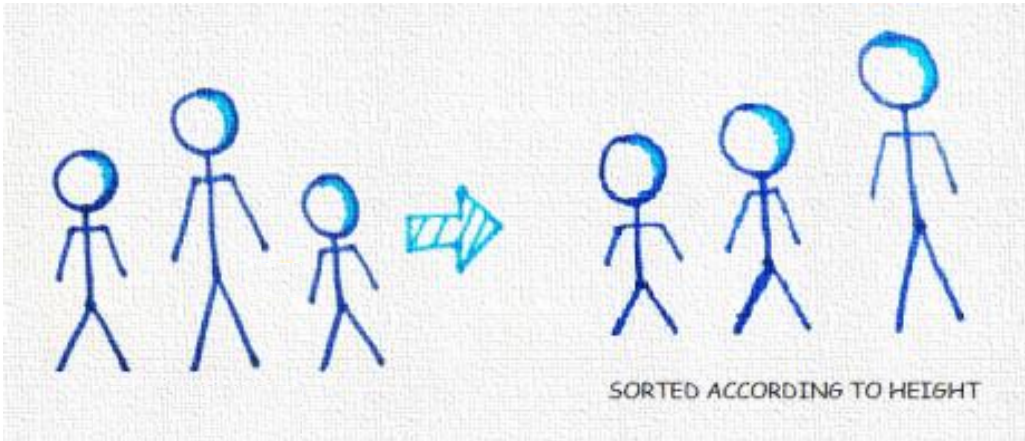


## Outline

- What is **Sorting Algorithm**?
- Key Characteristics of **Sorting Algorithms**
  - **Time Complexity**
  - **Space Complexity**
- **Bubble Sorting Algorithm**



# Concept of Sorting

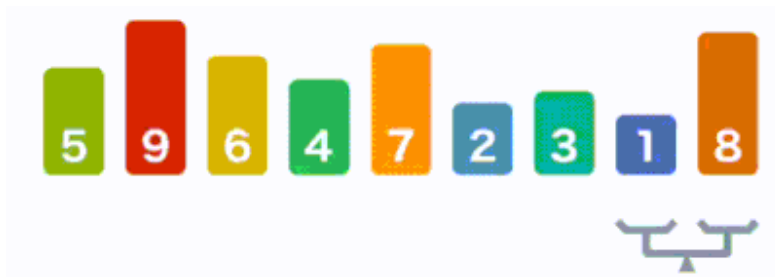


## Sorting Algorithms

- Sorting algorithms are the backbone of computer science and data processing, as they help organize and manage data efficiently.
- Sorting algorithms are widely used in many applications, **from search engines to e-commerce platforms.**
- Sorting algorithms are integral to ensuring smooth user experiences and optimized system performance.

# What is a Sorting Algorithm?

- A sorting algorithm is a procedure used to arrange elements of a list or dataset in a specific order, such as ascending or descending.



## Sorting Problem

Student records in a university.

Name	Grade	Group
Helin	2	B
Bana	3	A
Mohammed	2	A
Kawa	3	B
Diman	3	A

Sorting



Sorting records in ascending order.

Name	Grade	Group
Bana	3	A
Diman	3	A
Helin	2	B
Kawa	3	B
Mohammed	2	A

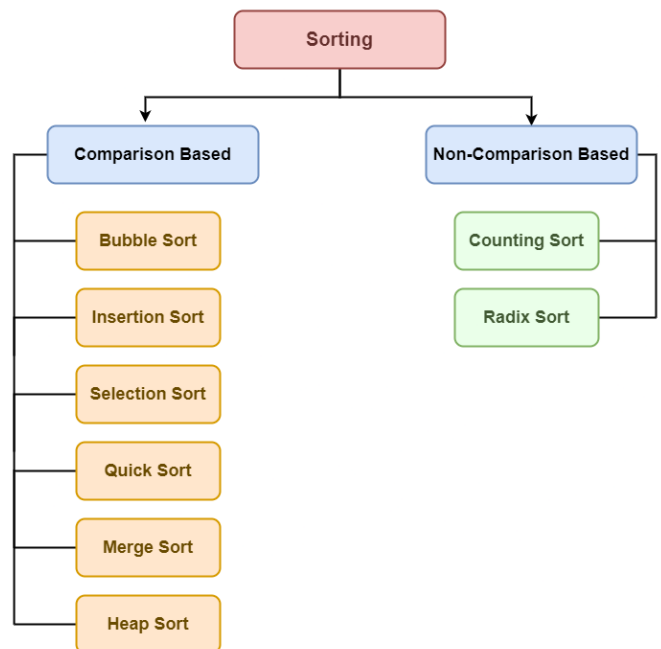


# Why Sorting Matters?

- Working on sorted data leads to **improve efficiency and performance.**
- Sorting algorithms are used to organize data for efficient **searching, retrieval, and analysis.**
- Sorting algorithms are generally classified into two types:
  - **Comparison-based** algorithms,
  - **Non-comparison-based** algorithms.

## Which Sorting Algorithm is the Best?

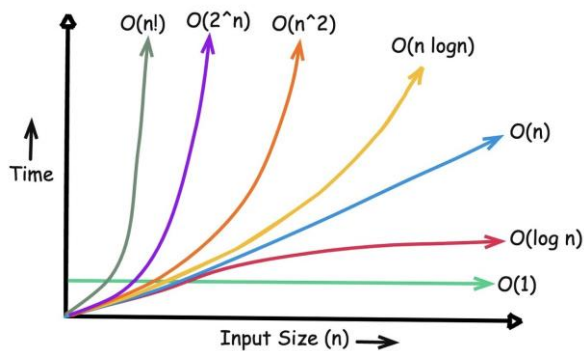
- Time Complexity
- Space Complexity





## Key Characteristics of Sorting Algorithms

- **Time Complexity:** The time an algorithm takes to sort a dataset, expressed in **Big O notation** (e.g.,  $O(n \log n)$ ,  $O(n^2)$ ).



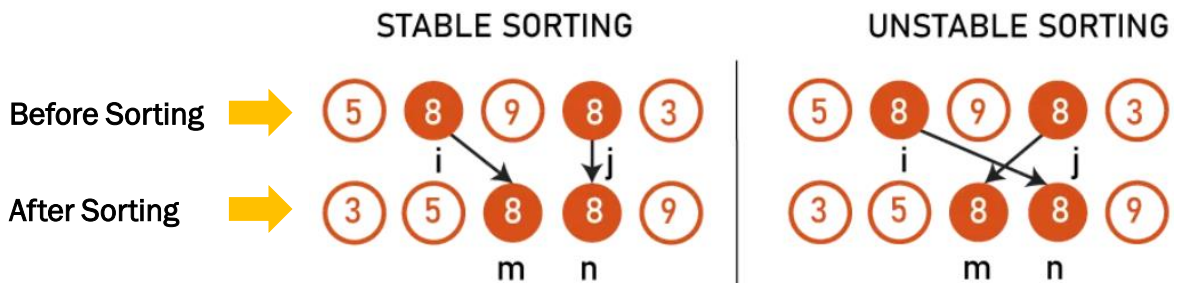
## Key Characteristics of Sorting Algorithms

- **Space Complexity:** Indicates the amount of extra memory used by the algorithm.
  - **In-Place Sorting:** These algorithms sort the data without needing extra space.
  - **Out-of-Place Sorting:** These algorithms need additional memory for temporary storage.



# Key Characteristics of Sorting Algorithms

- **Stability:** A sorting algorithm is considered stable if it does not change the order of elements with the same value.



# Time Complexity & Space Complexity of Some Sorting Algorithms



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	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$



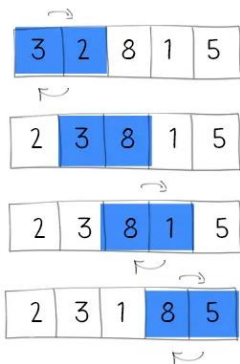
# Bubble Sort

- **Bubble Sort** repeatedly goes through a list, compares each pair of adjacent elements and swaps them if they are in the wrong order.
- This process continues until no more swaps are needed, meaning the list is sorted.
- With every new round, the largest element in the list “bubbles up” toward its correct position (end of list).

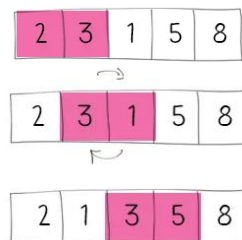


## How Bubble Sort Works?

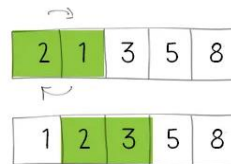
**Round 1**



**Round 2**



**Round 3**



**Round 4**





# How Bubble Sort Works?

Unsorted Data

3	2	8	1	5
---	---	---	---	---

Goal



Sorted Data

1	2	3	5	8
---	---	---	---	---

## Round 1

3	2	8	1	5
---	---	---	---	---

Swap

2	3	8	1	5
---	---	---	---	---

No Change!

2	3	8	1	5
---	---	---	---	---

Swap

2	3	1	8	5
---	---	---	---	---

Swap

At the end of round 1, the largest element in the list (8) "bubbles up" toward end of the list (its correct position).

2	3	1	5	8
---	---	---	---	---

Unsorted Data

3	2	8	1	5
---	---	---	---	---

Goal



Sorted Data

1	2	3	5	8
---	---	---	---	---

## Round 2

2	3	1	5	8
---	---	---	---	---

No Change!

2	3	1	5	8
---	---	---	---	---

Swap

2	1	3	5	8
---	---	---	---	---

No Change!

At the end of round 2, the second largest element in the list (5) "bubbles up" toward end of the list (its correct position).

2	1	3	5	8
---	---	---	---	---



# How Bubble Sort Works?



# How Bubble Sort Works?

Unsorted Data

3	2	8	1	5
---	---	---	---	---

Goal



Sorted Data

1	2	3	5	8
---	---	---	---	---

## Round 3

2	1	3	5	8
---	---	---	---	---

Swap

1	2	3	5	8
---	---	---	---	---

No Change!

At the end of round 3, the third largest element in the list (3) "bubbles up" toward end of the list (its correct position).

1	2	3	5	8
---	---	---	---	---

# How Bubble Sort Works?



Unsorted Data

3	2	8	1	5
---	---	---	---	---

Goal



Sorted Data

1	2	3	5	8
---	---	---	---	---

## Round 4

1	2	3	5	8
---	---	---	---	---

No Change!

In round 4, no swap happens at all. It means the data is sorted.

1	2	3	5	8
---	---	---	---	---

1	2	3	5	8
---	---	---	---	---





# When to Use Bubble Sort?

- Bubble Sort is considered as a simple algorithm to understand more complex sorting algorithms.
- **Bubble Sort** is a good sorting algorithm for:
  - **Small Datasets**
  - **Nearly Sorted Datasets**

# Bubble Sort Implementation in Python



```
##### Bubble Sort Function #####  
  
def bubbleSort(data) :  
    swapped = True  
    last = len(data) - 1  
    while(swapped) :  
        swapped = False  
        for i in range(0, last) :  
            if data[i] > data[i+1] :  
                swap(data, i, i+1)  
                swapped = True
```

```
##### Swapping two elements #####  
  
def swap(data, i, j) :  
    temp = data[i]  
    data[i] = data[j]  
    data[j] = temp
```



# Bubble Sort: Time and Space Complexity

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

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



# Thank You!