Bubble Sort Algorithm

Last Updated : 06 Aug, 2024

Bubble Sort is the simplest sorting algorithm 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 Bubble Sort 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.*

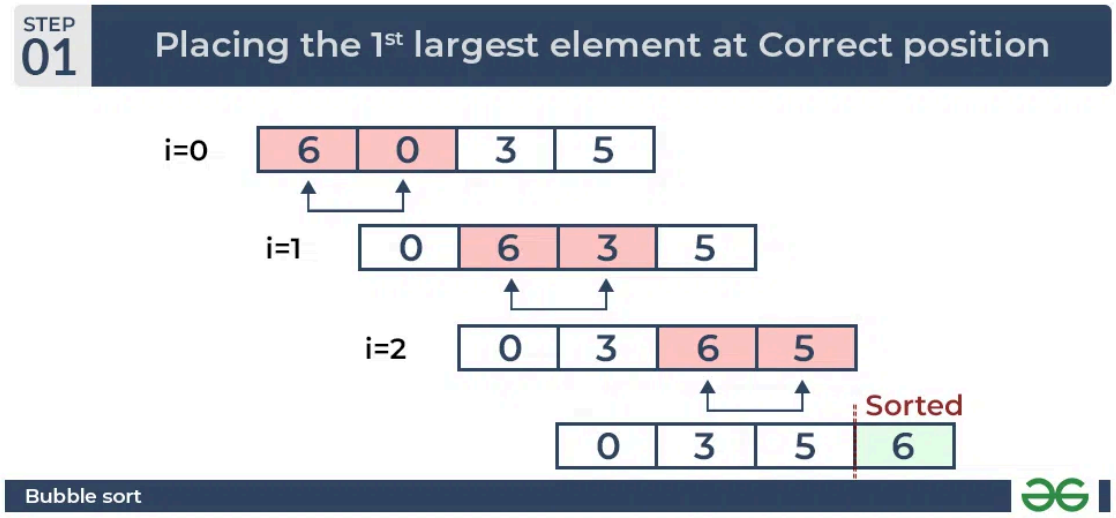
How does Bubble Sort Work?

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

Input: *arr[] = {6, 0, 3, 5}*

First Pass:

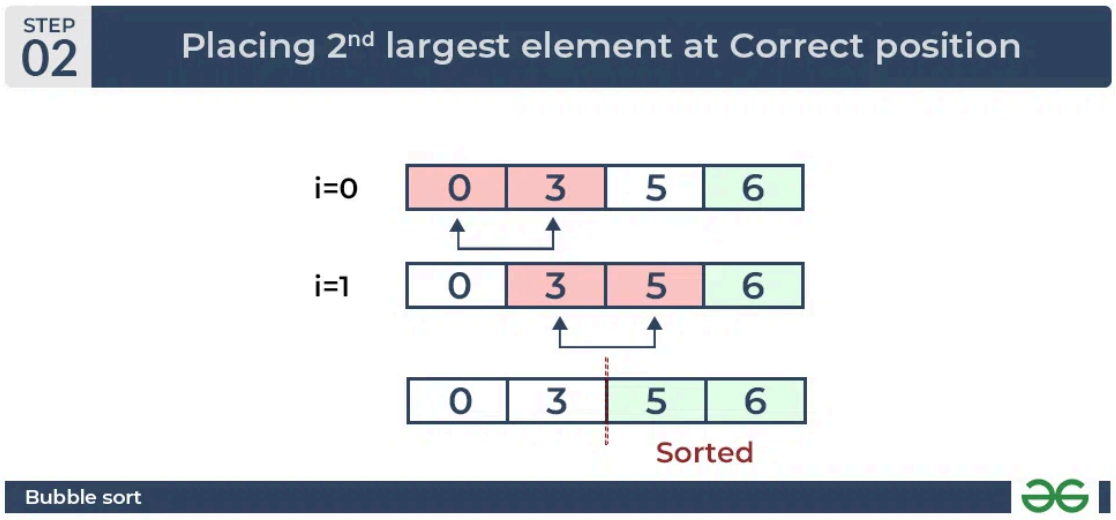
The largest element is placed in its correct position, i.e., the end of the array.



Bubble Sort Algorithm : Placing the largest element at correct position

Second Pass:

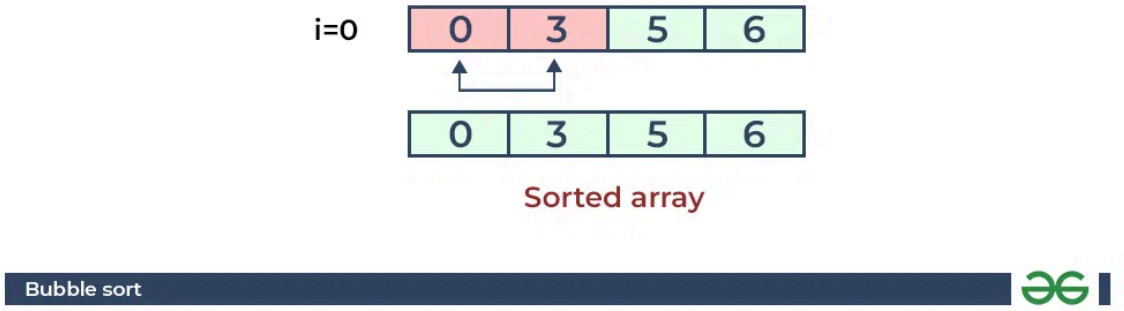
Place the second largest element at correct position



Bubble Sort Algorithm : Placing the second largest element at correct position

Third Pass:

Place the remaining two elements at their correct positions.



Bubble Sort Algorithm : Placing the remaining elements at their correct positions

- **Total no. of passes:** n-1
- **Total no. of comparisons:** n*(n-1)/2

Implementation of Bubble Sort

Below is the implementation of the bubble sort. It can be optimized by stopping the algorithm if the inner loop didn't cause any swap.

C++ C Java Python C# JavaScript PHP

```
// Optimized C# implementation of Bubble sort
using System;

class GFG {
    // An optimized version of Bubble Sort
    static void bubbleSort(int[] arr, int n)
    {
        int i, j, temp;
        bool swapped;
        for (i = 0; i < n - 1; i++) {
            swapped = false;
            for (j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {

                    // Swap arr[j] and arr[j+1]
                    temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                    swapped = true;
                }
            }
```

```
                // If no two elements were
                // swapped by inner loop, then break
                if (swapped == false)
                    break;
            }
        }

        // Function to print an array
        static void printArray(int[] arr, int size)
        {
            int i;
            for (i = 0; i < size; i++)
                Console.Write(arr[i] + " ");
            Console.WriteLine();
        }

        // Driver method
        public static void Main()
        {
            int[] arr = { 64, 34, 25, 12, 22, 11, 90 };
            int n = arr.Length;
            bubbleSort(arr, n);
            Console.WriteLine("Sorted array:");
            printArray(arr, n);
        }
    }
    // This code is contributed by Sam007
```

Output

```
Sorted array:
 11 12 22 25 34 64 90
```

Complexity Analysis of Bubble Sort:

Time Complexity: O(N^2)

Auxiliary Space: O(1)

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 $O(N^2)$ 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.

Some FAQs related to Bubble Sort:

What is the Boundary Case for Bubble sort?

Bubble sort takes minimum time (Order of n) when elements are already sorted. Hence it is best to check if the array is already sorted or not beforehand, to avoid $O(N^2)$ time complexity.

Does sorting happen in place in Bubble sort?

Yes, Bubble sort performs the swapping of adjacent pairs without the use of any major data structure. Hence Bubble sort algorithm is an in-place algorithm.

Is the Bubble sort algorithm stable?

Yes, the bubble sort algorithm is stable.

Where is the Bubble sort algorithm used?

Due to its simplicity, bubble sort is often used to introduce the concept of a sorting algorithm. In computer graphics, it is popular for its capability to detect a tiny error (like a swap of just two elements) in almost-sorted arrays and fix it with just linear complexity (2n). It can also be used in special situations where swapping of only adjacent elements is allowed as it sorts the array by swapping only adjacent elements.

Example: It is used in a polygon filling algorithm, where bounding lines are sorted by their x coordinate at a specific scan line (a line parallel to the x-axis), and with incrementing y their order changes (two elements are swapped) only at intersections of two lines.

Related Articles:

- Recursive Bubble Sort
- Coding practice for sorting
- Quiz on Bubble Sort
- Complexity Analysis of Bubble Sort

Next Article

Recursive Bubble Sort

Similar Reads

Comparison among Bubble Sort, Selection Sort and Insertion Sort

Bubble Sort, Selection Sort, and Insertion Sort are simple sorting algorithms that are commonly used to sort small datasets or as building blocks for more comple...

15 min read

Bubble Sort algorithm using JavaScript

Bubble sort algorithm is an algorithm that sorts an array by comparing two adjacent elements and swapping them if they are not in the intended order. Her...

4 min read

Selection Sort VS Bubble Sort

Not a valid contributionIn this, we will cover the comparison between Selection Sort VS Bubble Sort. The resources required by Selection Sort & Bubble So...

13 min read

Sort an array using Bubble Sort without using loops

Given an array arr[] consisting of N integers, the task is to sort the given array by using Bubble Sort without using loops. Examples: Input: arr[] = {1, 3, 4, 2,...

9 min read

Is Comb Sort better than Bubble Sort?

Comb sort and bubble sort are both simple sorting algorithms that are easy to implement. However, comb sort is generally considered to be more efficient tha...

2 min read

Bubble Sort On Doubly Linked List

Sort the given doubly linked list using bubble sort. Examples: Input : 5 4 3 2 1 Output : 1 2 3 4 5 Input : 2 1 3 5 4 Output :1 2 3 4 5 Explanation: As we do in th...

8 min read

Visualizing Bubble sort using Python

Prerequisites: Introduction to Matplotlib, Introduction to PyQt5, Bubble Sort Learning any algorithm can be difficult, and since you are here at GeekforGeeks,...

3 min read

Bubble Sort Visualization using JavaScript

GUI(Graphical User Interface) helps in better understanding than programs. In this article, we will visualize Bubble Sort using JavaScript. We will see how the...

4 min read

Sorting Strings using Bubble Sort

Given an array of strings arr[]. Sort given strings using Bubble Sort and display the sorted array. In Bubble Sort, the two successive strings arr[i] and arr[i+1] are...

4 min read

Recursive Bubble Sort

Background : Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.Example:...

10 min read

Article Tags : DSA Sorting Algorithms-BubbleSort BubbleSort +1 More

Practice Tags : redBus Sorting

Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development

English Grammar
Commerce
World GK

Data Science
CS Subjects