



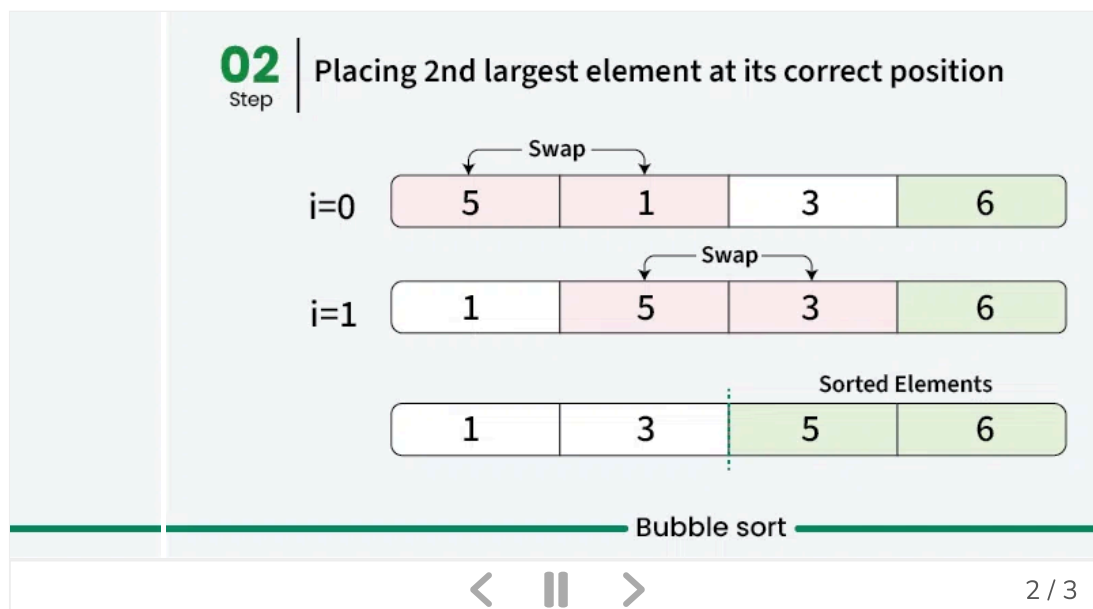
Bubble Sort Algorithm

Last Updated : 06 Oct, 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 are quite high.

- We sort the array using multiple passes. After the first pass, the maximum element goes to end (its correct position). Same way, after second pass, the second largest element goes to second last position and so on.
- In every pass, we process only those elements that have already not moved to correct position. After k passes, the largest k elements must have been moved to the last k positions.
- In a pass, we consider remaining elements and compare all adjacent and swap if larger element is before a smaller element. If we keep doing this, we get the largest (among the remaining elements) at its correct position.

How does Bubble Sort Work?



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](#)

```
1 # Optimized Python program for implementation of Bubble Sort
2 def bubbleSort(arr):
3     n = len(arr)
4
5     # Traverse through all array elements
6     for i in range(n):
7         swapped = False
8
9         # Last i elements are already in place
10        for j in range(0, n-i-1):
11
12            # Traverse the array from 0 to n-i-1
13            # Swap if the element found is greater
14            # than the next element
15            if arr[j] > arr[j+1]:
16                arr[j], arr[j+1] = arr[j+1], arr[j]
17                swapped = True
18        if (swapped == False):
19            break
20
21 # Driver code to test above
22 if __name__ == "__main__":
23     arr = [64, 34, 25, 12, 22, 11, 90]
24
25     bubbleSort(arr)
26
27     print("Sorted array:")
28     for i in range(len(arr)):
29         print("%d" % arr[i], end=" ")
```

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.

Frequently Asked Questions (FAQs) on 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.

Related Articles:

- [Recursive Bubble Sort](#)
- [Coding practice for sorting](#)
- [Quiz on Bubble Sort](#)
- [Complexity Analysis of Bubble Sort](#)

Join [GfG 160](#), a 160-day journey of coding challenges aimed at sharpening your skills. Each day, solve a handpicked problem, dive into detailed solutions through articles and videos, and enhance your preparation for any interview—all for free! Plus, win exciting GfG goodies along the way! - [Explore Now](#)



GeeksforGeeks



847

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 Sort...

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

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

Bubble sort using two Stacks

Prerequisite : Bubble Sort Write a function that sort an array of integers using stacks and also uses bubble sort paradigm. Algorithm: 1. Push all elements of...

6 min read

Article Tags :

DSA

Sorting

Algorithms-BubbleSort

BubbleSort

+1 More

Practice Tags :

redBus

Sorting



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



Company

- About Us
- Legal
- In Media
- Contact Us
- Advertise with us
- GFG Corporate Solution
- Placement Training Program
- GeeksforGeeks Community

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial
- Tutorials Archive

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation

[Top 100 DSA Interview Problems](#)[DSA Roadmap by Sandeep Jain](#)[All Cheat Sheets](#)[Pandas](#)[NumPy](#)[NLP](#)[Deep Learning](#)

Web Technologies

[HTML](#)[CSS](#)[JavaScript](#)[TypeScript](#)[ReactJS](#)[NextJS](#)[Bootstrap](#)[Web Design](#)

Computer Science

[Operating Systems](#)[Computer Network](#)[Database Management System](#)[Software Engineering](#)[Digital Logic Design](#)[Engineering Maths](#)[Software Development](#)[Software Testing](#)

System Design

[High Level Design](#)[Low Level Design](#)[UML Diagrams](#)[Interview Guide](#)[Design Patterns](#)[OOAD](#)[System Design Bootcamp](#)[Interview Questions](#)

School Subjects

[Mathematics](#)[Physics](#)[Chemistry](#)[Biology](#)[Social Science](#)[English Grammar](#)[Commerce](#)[World GK](#)

Python Tutorial

[Python Programming Examples](#)[Python Projects](#)[Python Tkinter](#)[Web Scraping](#)[OpenCV Tutorial](#)[Python Interview Question](#)[Django](#)

DevOps

[Git](#)[Linux](#)[AWS](#)[Docker](#)[Kubernetes](#)[Azure](#)[GCP](#)[DevOps Roadmap](#)

Interview Preparation

[Competitive Programming](#)[Top DS or Algo for CP](#)[Company-Wise Recruitment Process](#)[Company-Wise Preparation](#)[Aptitude Preparation](#)[Puzzles](#)

GeeksforGeeks Videos

[DSA](#)[Python](#)[Java](#)[C++](#)[Web Development](#)[Data Science](#)[CS Subjects](#)