

Bubble Sort

Difficulty Level : Easy • Last Updated : 09 Mar, 2022

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

Example:

First Pass:

(**5** 1 4 2 8) -> (**1** **5** 4 2 8), Here, algorithm compares the first two elements, and swaps since $5 > 1$.

(1 **5** 4 2 8) -> (1 **4** **5** 2 8), Swap since $5 > 4$

(1 4 **5** 2 8) -> (1 4 **2** **5** 8), Swap since $5 > 2$

(1 4 2 **5** 8) -> (1 4 2 **5** 8), Now, since these elements are already in order ($8 > 5$), algorithm does not swap them.

Second Pass:

(**1** **4** 2 5 8) -> (**1** **4** 2 5 8)

(1 **4** **2** 5 8) -> (1 **2** **4** 5 8), Swap since $4 > 2$

(1 2 **4** **5** 8) -> (1 2 **4** **5** 8)

(1 2 4 **5** 8) -> (1 2 4 **5** 8)

Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Third Pass:

(**1** **2** 4 5 8) -> (**1** **2** 4 5 8)

(1 **2** **4** 5 8) -> (1 **2** **4** 5 8)

(1 2 **4** **5** 8) -> (1 2 **4** **5** 8)

(1 2 4 **5** 8) -> (1 2 4 **5** 8)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

Data Structures & Algorithms Self-Paced Course

[Enrol Now](#)

Recommended: Please solve it on "**PRACTICE**" first, before moving on to the solution.

Following is the implementations of Bubble Sort.

C++

```
// C++ program for implementation of Bubble sort
#include <bits/stdc++.h>
using namespace std;

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

// A function to implement bubble sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)

        // Last i elements are already in place
        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}

/* Function to print an array */
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    cout<<"Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```

// This code is contributed by rathbhupendra

C

// C program for implementation of Bubble sort
#include <stdio.h>

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

// A function to implement bubble sort

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)

        // Last i elements are already in place
        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}
```

/* Function to print an array */

```
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
printArray(arr, n);  
return 0;  
}
```

Java

```
// Java program for implementation of Bubble Sort  
class BubbleSort  
{  
    void bubbleSort(int arr[])  
    {  
        int n = arr.length;  
        for (int i = 0; i < n-1; i++)  
            for (int j = 0; j < n-i-1; j++)  
                if (arr[j] > arr[j+1])  
                {  
                    // swap arr[j+1] and arr[j]  
                    int temp = arr[j];  
                    arr[j] = arr[j+1];  
                    arr[j+1] = temp;  
                }  
    }  
  
    /* Prints the array */  
    void printArray(int arr[])  
    {  
        int n = arr.length;  
        for (int i=0; i<n; ++i)  
            System.out.print(arr[i] + " ");  
        System.out.println();  
    }  
  
    // Driver method to test above  
    public static void main(String args[])  
    {  
        BubbleSort ob = new BubbleSort();  
        int arr[] = {64, 34, 25, 12, 22, 11, 90};  
        ob.bubbleSort(arr);  
        System.out.println("Sorted array");  
        ob.printArray(arr);  
    }  
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```

def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):

        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i],end=" ")

```

C#

```

// C# program for implementation
// of Bubble Sort
using System;

class GFG
{
    static void bubbleSort(int []arr)
    {
        int n = arr.Length;
        for (int i = 0; i < n - 1; i++)
            for (int j = 0; j < n - i - 1; j++)
                if (arr[j] > arr[j + 1])
                {
                    // swap temp and arr[i]
                    int temp = arr[j];

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```

        Console.Write(arr[i] + " ");
        Console.WriteLine();
    }

    // Driver method
    public static void Main()
    {
        int []arr = {64, 34, 25, 12, 22, 11, 90};
        bubbleSort(arr);
        Console.WriteLine("Sorted array");
        printArray(arr);
    }
}

// This code is contributed by Sam007

```

PHP

```

<?php
// PHP program for implementation
// of Bubble Sort

function bubbleSort(&$arr)
{
    $n = sizeof($arr);

    // Traverse through all array elements
    for($i = 0; $i < $n; $i++)
    {
        // Last i elements are already in place
        for ($j = 0; $j < $n - $i - 1; $j++)
        {
            // traverse the array from 0 to n-i-1
            // Swap if the element found is greater
            // than the next element
            if ($arr[$j] > $arr[$j+1])
            {
                $t = $arr[$j];
                $arr[$j] = $arr[$j+1];
            }
        }
    }
}

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
echo "Sorted array : \n";

for ($i = 0; $i < $len; $i++)
    echo $arr[$i]. " ";

// This code is contributed by ChitraNayal.
?>
```

Javascript

```
<script>

function swap(arr, xp, yp)
{
    var temp = arr[xp];
    arr[xp] = arr[yp];
    arr[yp] = temp;
}

// An optimized version of Bubble Sort
function bubbleSort( arr, n)
{
    var i, j;
    for (i = 0; i < n-1; i++)
    {
        for (j = 0; j < n-i-1; j++)
        {
            if (arr[j] > arr[j+1])
            {
                swap(arr, j, j+1);
            }
        }
    }
}

/* Function to print an array */
function printArray(arr, size)
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```
printArray(arr, n);

bubbleSort(arr, n);
document.write("Sorted array: \n");
printArray(arr, n);
```

```
</script>
```

Output:

Sorted array:

11 12 22 25 34 64 90

<!--Illustration :

| | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|---|
| i = 0 | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | | 5 | 3 | 1 | 9 | 8 | 2 | 4 | 7 |
| 1 | | 3 | 5 | 1 | 9 | 8 | 2 | 4 | 7 |
| 2 | | 3 | 1 | 5 | 9 | 8 | 2 | 4 | 7 |
| 3 | | 3 | 1 | 5 | 9 | 8 | 2 | 4 | 7 |
| 4 | | 3 | 1 | 5 | 8 | 9 | 2 | 4 | 7 |
| 5 | | 3 | 1 | 5 | 8 | 2 | 9 | 4 | 7 |
| 6 | | 3 | 1 | 5 | 8 | 2 | 4 | 9 | 7 |
| i = 1 | 0 | 3 | 1 | 5 | 8 | 2 | 4 | 7 | 9 |
| 1 | | 1 | 3 | 5 | 8 | 2 | 4 | 7 | |
| 2 | | 1 | 3 | 5 | 8 | 2 | 4 | 7 | |
| 3 | | 1 | 3 | 5 | 8 | 2 | 4 | 7 | |
| 4 | | 1 | 3 | 5 | 2 | 8 | 4 | 7 | |
| 5 | | 1 | 3 | 5 | 2 | 4 | 8 | 7 | |
| i = 2 | 0 | 1 | 3 | 5 | 2 | 4 | 7 | 8 | |
| 1 | | 1 | 3 | 5 | 2 | 4 | 7 | | |
| 2 | | 1 | 3 | 5 | 2 | 4 | 7 | | |
| 3 | | 1 | 3 | 2 | 5 | 4 | 7 | | |
| 4 | | 1 | 3 | 2 | 4 | 5 | 7 | | |
| i = 3 | 0 | 1 | 3 | 2 | 4 | 5 | 7 | | |
| 1 | | 1 | 3 | 2 | 4 | 5 | | | |
| 2 | | 1 | 2 | 3 | 4 | 5 | | | |
| 3 | | 1 | 2 | 3 | 4 | 5 | | | |
| i = 4 | 0 | 1 | 2 | 3 | 4 | 5 | | | |
| 1 | | 1 | 2 | 3 | 4 | | | | |
| 2 | | 1 | 2 | 3 | 4 | | | | |
| i = 5 | 0 | 1 | 2 | 3 | 4 | | | | |
| 1 | | 1 | 2 | 3 | | | | | |
| i = 6 | 0 | 1 | 2 | 3 | | | | | |
| | 1 | 2 | 3 | | | | | | |

—>

Optimized Implementation:

The above function always runs $O(n^2)$ time even if the array is sorted. It can be

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
using namespace std;
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

// An optimized version of Bubble Sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    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], &arr[j+1]);
                swapped = true;
            }
        }

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

/* Function to print an array */
void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        cout << " " << arr[i];
    cout << " n";
}

// Driver program to test above functions
int main()
{
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

C



```
// Optimized implementation of Bubble sort
#include <stdio.h>
#include <stdbool.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

// An optimized version of Bubble Sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    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], &arr[j+1]);
                swapped = true;
            }
        }

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

/* Function to print an array */
void printArray(int arr[], int size)
{

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
bubbleSort(arr, n);  
printf("Sorted array: \n");  
printArray(arr, n);  
return 0;  
}
```

Java

```
// Optimized java implementation  
// of Bubble sort  
import java.io.*;  
  
class GFG  
{  
    // An optimized version of Bubble Sort  
    static void bubbleSort(int arr[], int n)  
    {  
        int i, j, temp;  
        boolean 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;  
        }  
    }  
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
public static void main(String args[])
{
    int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
    int n = arr.length;
    bubbleSort(arr, n);
    System.out.println("Sorted array: ");
    printArray(arr, n);
}
}
```

// This code is contributed
// by Nikita Tiwari.

Python3

```
# Python3 Optimized implementation
# of Bubble sort

# An optimized version of Bubble Sort
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):
        swapped = False

        # Last i elements are already
        # in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to
            # n-i-1. Swap if the element
            # found is greater than the
            # next element
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True

    # IF no two elements were swapped
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
print ("%d" %arr[i],end=" ")
```

This code is contributed by Shreyanshi Arun

C#

```
// 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
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```

    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

```

PHP

```

<?php
// PHP Optimized implementation
// of Bubble sort

// An optimized version of Bubble Sort
function bubbleSort(&$arr)
{
    $n = sizeof($arr);

    // Traverse through all array elements
    for($i = 0; $i < $n; $i++)
    {
        $swapped = False;

        // Last i elements are already
        // in place
        for ($j = 0; $j < $n - $i - 1; $j++)
        {

            // traverse the array from 0 to
            // n-i-1. Swap if the element
            // found is greater than the
            // next element
            if ($arr[$j] > $arr[$j+1])
            {
                $t = $arr[$j];
                $arr[$j] = $arr[$j+1];
                $arr[$j+1] = $t;
            }
        }
    }
}

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)
[Register](#)

```
// Driver code to test above
$arr = array(64, 34, 25, 12, 22, 11, 90);
$len = sizeof($arr);
bubbleSort($arr);

echo "Sorted array : \n";

for($i = 0; $i < $len; $i++)
    echo $arr[$i]. " ";

// This code is contributed by ChitraNayal.
?>
```

Javascript

```
<script>

// Optimized javaScript implementation
// of Bubble sort
// An optimized version of Bubble Sort
function bubbleSort(arr, n)
{
    var i, j, temp;
    var 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
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

```
        document.write(arr[i] + " ");
        document.writeln();
    }

    // Driver program
    var arr = [ 64, 34, 25, 12, 22, 11, 90 ];
    var n = arr.length;
    bubbleSort(arr, n);
    document.write("Sorted array: ");
    printArray(arr, n);

    // This code is contributed shivanisinghss2110
</script>
```

Output:

Sorted array:

11 12 22 25 34 64 90

Worst and Average Case Time Complexity: $O(n^2)$. Worst case occurs when array is reverse sorted.

Best Case Time Complexity: $O(n)$. Best case occurs when array is already sorted.

Auxiliary Space: $O(1)$

Boundary Cases: Bubble sort takes minimum time (Order of n) when elements are already sorted.

Sorting In Place: Yes

Stable: Yes

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 very small error (like

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

Bubble Sort | GeeksforGeeks



?list=PLqM7aHXFySHrGIxeBOo4-mKO4H8j2knW

Snapshots:

[Quiz on Bubble Sort](#)

Other Sorting Algorithms on GeeksforGeeks/GeeksQuiz:

[Recursive Bubble Sort](#)

[Coding practice for sorting.](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

RECOMMENDED ARTICLES

Page : **1** 2 3

01 Comparison among Bubble Sort, Selection Sort and Insertion Sort
01, Apr 19

05 Bubble sort using two Stacks
22, Jun 17

02 Sort an array using Bubble Sort without using loops
04, Mar 21

06 C Program for Bubble Sort
02, Feb 14

03 Selection Sort VS Bubble Sort
14, Dec 20

07 C++ Program for Bubble Sort
02, Feb 14

04 Recursive Bubble Sort
15, Feb 17

08 Java Program for Bubble Sort
02, Feb 14

Article Contributed By :



GeeksforGeeks

Vote for difficulty

Current difficulty : [Easy](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)[Improve Article](#)[Report Issue](#)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

- [About Us](#)
- [Careers](#)
- [In Media](#)
- [Contact Us](#)
- [Privacy Policy](#)
- [Copyright Policy](#)

Learn

- [Algorithms](#)
- [Data Structures](#)
- [SDE Cheat Sheet](#)
- [Machine learning](#)
- [CS Subjects](#)
- [Video Tutorials](#)

News

Languages

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Start Your Coding Journey Now!

[Login](#)[Register](#)

Web Development

[Web Tutorials](#)[Django Tutorial](#)[HTML](#)[CSS](#)[JavaScript](#)[Bootstrap](#)

Contribute

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Write Interview Experience](#)[Internships](#)[Video Internship](#)

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !