# 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:**

**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** )

Following is iterative Bubble sort algorithm :

```
// Iterative Bubble Sort
bubbleSort(arr[], n)
{
  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]);
}
```

See Bubble Sort for more details.

**How to implement it recursively?**

Recursive Bubble Sort has no performance/implementation advantages, but can be a good question to check one's understanding of Bubble Sort and recursion.

If we take a closer look at Bubble Sort algorithm, we can notice that in first pass, we move largest element to end (Assuming sorting in increasing order). In second pass, we move second largest element to second last position and so on.

**Recursion Idea.**

1. Base Case: If array size is 1, return.
2. Do One Pass of normal Bubble Sort. This pass fixes last element of current subarray.
3. Recur for all elements except last of current subarray.

Below is implementation of above idea.

---

## C/C++

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

// A function to implement bubble sort
void bubbleSort(int arr[], int n)
{
    // Base case
    if (n == 1)
        return;

    // One pass of bubble sort. After
    // this pass, the largest element
    // is moved (or bubbled) to end.
    for (int i=0; i<n-1; i++)
        if (arr[i] > arr[i+1])
```

```
            swap(arr[i], arr[i+1]);

    // Largest element is fixed,
    // recur for remaining array
    bubbleSort(arr, n-1);
}

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

// Driver program to test above functions
int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    printf("Sorted array : \n");
    printArray(arr, n);
    return 0;
}
```

Run on IDE

## Java

```
// Java program for recursive implementation
// of Bubble sort

import java.util.Arrays;

public class GFG
{
    // A function to implement bubble sort
    static void bubbleSort(int arr[], int n)
    {
        // Base case
        if (n == 1)
            return;

        // One pass of bubble sort. After
        // this pass, the largest element
        // is moved (or bubbled) to end.
        for (int i=0; i<n-1; i++)
            if (arr[i] > arr[i+1])
            {
                // swap arr[i], arr[i+1]
                int temp = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = temp;
            }

        // Largest element is fixed,
        // recur for remaining array
        bubbleSort(arr, n-1);
    }

    // Driver Method
    public static void main(String[] args)
    {
        int arr[] = {64, 34, 25, 12, 22, 11, 90};

        bubbleSort(arr, arr.length);
```

```
        System.out.println("Sorted array : ");
        System.out.println(Arrays.toString(arr));
    }
}
```

Run on IDE

## Python3

```
# Python Program for implementation of
# Recursive Bubble sort

def bubble_sort(listt):
    for i, num in enumerate(listt):
        try:
            if listt[i+1] < num:
                listt[i] = listt[i+1]
                listt[i+1] = num
                bubble_sort(listt)
        except IndexError:
            pass
    return listt

listt = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(listt)

print("Sorted array:");
for i in range(0, len(listt)):
    print(listt[i], end=' ')


# Code contributed by Mohit Gupta_OMG
```

Run on IDE

**Output :**

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

This article is contributed by **Suprotik Dey**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**Practice Tags :** Recursion Sorting

**Article Tags :** Recursion Sorting

Login to Improve this Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

## Recommended Posts:

Insertion Sort
Bubble Sort
Recursive Insertion Sort
Selection Sort
Merge Sort
Find n-th node in Postorder traversal of a Binary Tree
Convert a Binary Tree such that every node stores the sum of all nodes in its right subtree
Recursive program to generate power set
Check for balanced parenthesis without using stack
Smallest number in BST which is greater than or equal to N

(Login to Rate)

1.6 Average Difficulty : **1.6/5.0**
Based on **35** vote(s)

Basic Easy Medium Hard Expert

Add to TODO List
Mark as DONE

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

Share this post!

**19 Comments** **GeeksforGeeks** **1 Login**

**Recommend** **Share** **Sort by Newest**

Join the discussion…

**LOG IN WITH** **OR SIGN UP WITH DISQUS** ?

Name

**Anandesh Sharma** • 19 days ago

*Proper Recursive Bubble Sort implementation in Java*

```
public class RecursiveBS{
        public static void sort(int[] A, int n, int i, int j){
                int swap = 0;
                if(i >= n-1)
                        return;
                if(j < n - 1){
                        sort(A, n, i, j + 1);
                        if(A[j] > A[j+1]){
                                swap = A[j];
                                A[j] = A[j+1];
                                A[j+1] = swap;
                        }
                }
                if(j == 0)
                        sort(A, n, i + 1, 0);
        }

        public static void main(String...args){
                int[] A = {0,9,8,7,6,5,4,3,2,1};
                sort(A, A.length, 0, 0);
                for(int i = 0; i < A.length; i++){
                        System.out.print(A[i] + " ");
                }
        }
}
```



**Ali Khalilli** • 22 days ago

```
def bubbleSort(arr, c_index):
    if c_index == 0:
        return
    bubbleSort(arr, c_index-1)
    j = len(arr) - c_index - 1
    while j < len(arr)-1:
        if arr[j] > arr[j+1]:
            arr[j], arr[j+1] = arr[j+1], arr[j]
        j += 1
```



**Ravisankar** • a month ago

```
public void recusiveBubbleSortSwap(int[] array,int index,int length) {
if (index< length) {
if (array[index]> array[index+1]) {
int temp = array[index];
array[index] = array[index+1];
array[index+1] = temp;
}
recusiveBubbleSortSwap(array, index+1, length);
}
return ;
}

public void recursiveBubbleSort(int[] array, int index,int length) {
if (index <= length) {
recusiveBubbleSortSwap(array, 0, length--);
recursiveBubbleSort(array, index+1, length);
}
}

}
```

}

Reply • Share ›

**Rohitashwa Nigam** • 2 months ago

The python implementation looks to be O(n^3), rather than O(n^2).!
The C/C++ implementation is correct.
In the python implementation, to move the largest element to its place, in the first iteration we would have taken O(n^2).

Reply • Share ›

**Rohitashwa Nigam** ➚ Rohitashwa Nigam • 2 months ago

```
# this will do the job
from array import array

def recursiveBubbleSort(arr):

swapped = False
for i in range(0, len(arr)-1):
if arr[i] > arr[i+1]:
arr[i], arr[i+1] = arr[i+1], arr[i]
swapped = True

if not swapped:
return

recursiveBubbleSort(arr)

def printArray(arr):
for i in range(0, len(arr)):
print(arr[i], end = ' ')

if __name__ == '__main__':
arr = array('i', [int(x) for x in input('Enter the array ').split(',')])
recursiveBubbleSort(arr)
printArray(arr)
```

Reply • Share ›

**Rohitashwa Nigam** ➚ Rohitashwa Nigam • 2 months ago

Sorry for the indentations.!!

Reply • Share ›

**Harshit Chawla** • 5 months ago

```
public static void Rec2bubbleSort(int []arr, int bi, int ei)
{
if(ei==0)
{
return;
}
```

```
if(bi==ei-1)
{
Rec2bubbleSort(arr, 0, ei-1);
return ;
}
if(arr[bi]>arr[bi+1])
{
int temp=arr[bi];
arr[bi]=arr[bi+1];
arr[bi+1]=temp;
}
Rec2bubbleSort(arr, bi+1, ei);
}
```

^ | v • Reply • Share ›

**Cengiz Üçgül** • 7 months ago

I love this web site

^ | v • Reply • Share ›

**Ronny S** • 7 months ago

Totally recursive (no loops):
https://ide.geeksforgeeks.o...

^ | v • Reply • Share ›

**vikas yadav** • 10 months ago

http://ide.geeksforgeeks.or... this is true Recursion not that is mentioned above

^ | v • Reply • Share ›

**Suprotik Dey** ➚ vikas yadav • 2 months ago

recursion does not say that a function will have no loops

^ | v • Reply • Share ›

**vikas yadav** ➚ vikas yadav • 10 months ago

http://ide.geeksforgeeks.or... this is correct

^ | v • Reply • Share ›

**Hitesh Kumar** ➚ vikas yadav • 5 months ago

It does not sort input array [2, 1, 3, 5, 4] correctly. Output is: [1 2 3 5 4 ]

^ | v • Reply • Share ›

**Sand Summerstorm** • a year ago

Why is it incorrect

swap(&arr[i], &arr[i+1]);

?

And please geeks explain to me what exactly return does in this case.

Reply • Share ›

**Suprotik Dey** ➚ Sand Summerstorm • 8 months ago

Mind you are not in c! its c++ and you are using stl function.

Reply • Share ›

**harsh vardhan** ➚ Sand Summerstorm • a year ago

swap is a pre defined function in stl which accepts the values as parameters and not their addresses

Reply • Share ›

**Vivek Ramesh** • a year ago

```
public static void sort(int[] arr, int i,int j){
if(i!=arr.length){
if(j==arr.length){
sort(arr,i+1,0);
}
else{
if(j<arr.length-i-1 &&arr[j]="">arr[j+1]){
int temp=arr[j+1];
arr[j+1]=arr[j];
arr[j]=temp;
}
sort(arr,i,j+1);
}
}
}
```

Reply • Share ›

**Vivek Ramesh** • a year ago

Below java code without using any loop-

```
public static void sort(int[] arr, int i,int j){
if(i!=arr.length){
if(j==arr.length){
sort(arr,i+1,0);
}
else{
if(j<arr.length-i-1 &&arr[j]="">arr[j+1]){
int temp=arr[j+1];
arr[j+1]=arr[j];
arr[j]=temp;
}
sort(arr,i,j+1);
}
```

```
}
}
```

Reply • Share ›

**gopal** • a year ago

```
//code without any loop(fully recursive code)
#include <stdio.h>
int j=0;
void Rec2(int arr[],int j,int n){
if(arr[j]>arr[j+1]){
arr[j]=arr[j]+arr[j+1];
arr[j+1]=arr[j]-arr[j+1];
arr[j]=arr[j]-arr[j+1];
}
if(++j<n){ rec2(arr,j,n);="" }="" }="" void="" rec1(int="" arr[],int="" n){="" if(n="0)" return;="" rec2(arr,0,n);="" rec1(arr,n-1);="" }="" int="" main(void)="" {="" your="" code="" goes="" here="" int="" arr[5]="{5,1,4,2,8};" int="" n="sizeof(arr)/sizeof(arr[0]);" int="" i="0;" rec1(arr,n-1);="" for(i="0;i&amp;lt;n;i++){" printf("%d="" ",arr[i]);="" }="" return="" 0;="" }="" link="" :="" https:="" ideone.com="" xhod5h="">
```

Reply • Share ›