





🗐 Jobs 💰 Examples 🖪 Qries 🛂 Whiteboard 💌 Net Meeting 🧔 Tools 🖹 Articles



Tutorials Point (India) Pvt. Ltd.

► YouTube 65K



LEARN DATA STRUCTURE

organizing data

Data Structures & Algorithms

DSA - Home

DSA - Overview

DSA - Environment Setup

Algorithm

DSA - Algorithms Basics

DSA - Asymptotic Analysis

DSA - Greedy Algorithms

DSA - Divide and Conquer

DSA - Dynamic Programming

Data Structures

DSA - Data Structure Basics

DSA - Array Data Structure

Linked Lists

DSA - Linked List Basics

DSA - Doubly Linked List

DSA - Circular Linked List

Stack & Queue

DSA - Stack

DSA - Expression Parsing

DSA - Queue

Searching Techniques

DSA - Binary Search

DSA - Linear Search

DSA - Interpolation Search

DSA - Hash Table

Sorting Techniques

DSA - Sorting Algorithms

DSA - Bubble Sort

DSA - Insertion Sort

DSA - Selection Sort

DSA - Merge Sort DSA - Shell Sort

DSA - Quick Sort

DSA - Graph Data Structure

Graph Data Structure

DSA - Depth First Traversal

DSA - Breadth First Traversal

Tree Data Structure

DSA - Tree Data Structure DSA - Tree Traversal

DSA - Binary Search Tree

DSA - AVL Tree DSA - Spanning Tree

DSA - Heap

DSA - Recursion Basics

Recursion

DSA - Tower of Hanoi

DSA - Fibonacci Series

DSA Useful Resources

DSA - Questions and Answers DSA - Quick Guide

DSA - Useful Resources DSA - Discussion

Selected Reading

Developer's Best Practices

UPSC IAS Exams Notes

Questions and Answers Effective Resume Writing

HR Interview Questions Computer Glossary

Who is Who

Data Structure - Bubble Sort Algorithm

Advertisements

Dashboard Tool Free Trial

BUILD YOUR FIRST DASHBOARD TODAY

Free Trial >

Previous Page

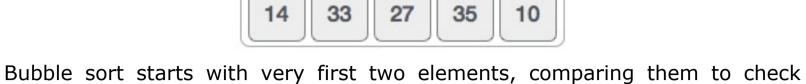
which one is greater.

Next Page **⊙**

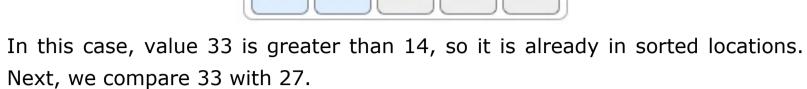
Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of $O(n^2)$ where **n** is the number of items.

How Bubble Sort Works?

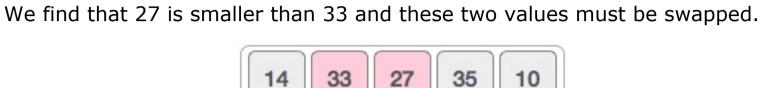
we're keeping it short and precise.



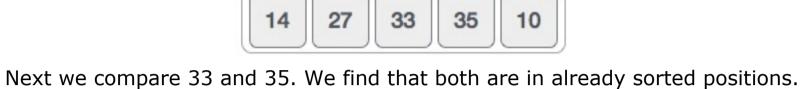
33 27 35 10



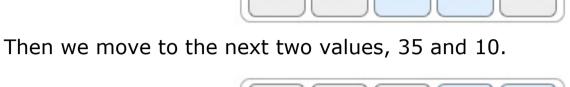
33 27 35 10 14



The new array should look like this -



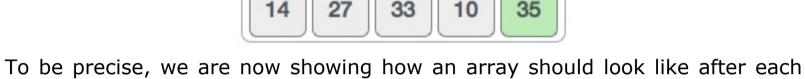
27 33 35 10



33 35 27 10



After one iteration, the array should look like this -



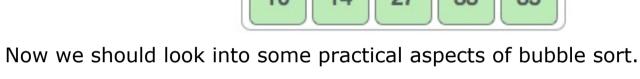
10 35

iteration. After the second iteration, it should look like this -

Notice that after each iteration, at least one value moves at the end.



27 35



We assume **list** is an array of **n** elements. We further assume that **swap**

function swaps the values of the given array elements.

if list[i] > list[i+1]

Algorithm

completely sorted.

begin BubbleSort(list) for all elements of list

```
swap(list[i], list[i+1])
      end if
   end for
   return list
end BubbleSort
Pseudocode
```

We observe in algorithm that Bubble Sort compares each pair of array element unless the whole array is completely sorted in an ascending order. This may

loop = list.count;

as all the elements are already ascending. To ease-out the issue, we use one flag variable **swapped** which will help us see if any swap has happened or not. If no swap has occurred, i.e. the array requires no more processing to be sorted, it will come out of the loop.

cause a few complexity issues like what if the array needs no more swapping

Pseudocode of BubbleSort algorithm can be written as follows procedure bubbleSort(list : array of items)

for i = 0 to loop-1 do: swapped = false

```
for j = 0 to loop-1 do:
         /* compare the adjacent elements */
         if list[j] > list[j+1] then
            /* swap them */
            swap( list[j], list[j+1] )
            swapped = true
         end if
      end for
       /*if no number was swapped that means
       array is sorted now, break the loop.*/
      if(not swapped) then
         break
      end if
   end for
end procedure return list
Implementation
One more issue we did not address in our original algorithm and its improvised
pseudocode, is that, after every iteration the highest values settles down at
the end of the array. Hence, the next iteration need not include already sorted
```

elements. For this purpose, in our implementation, we restrict the inner loop

to avoid already sorted values. To know about bubble sort implementation in C programming language, please click here ... Print • Previous Page Next Page **⊙** Advertisements



© Copyright 2018. All Rights Reserved.

