

[Write an Article](#)[Login](#)

## Insertion Sort

Insertion sort is a simple sorting algorithm that works the way we sort playing cards in our hands.

**2.1**

### Algorithm

// Sort an arr[] of size n

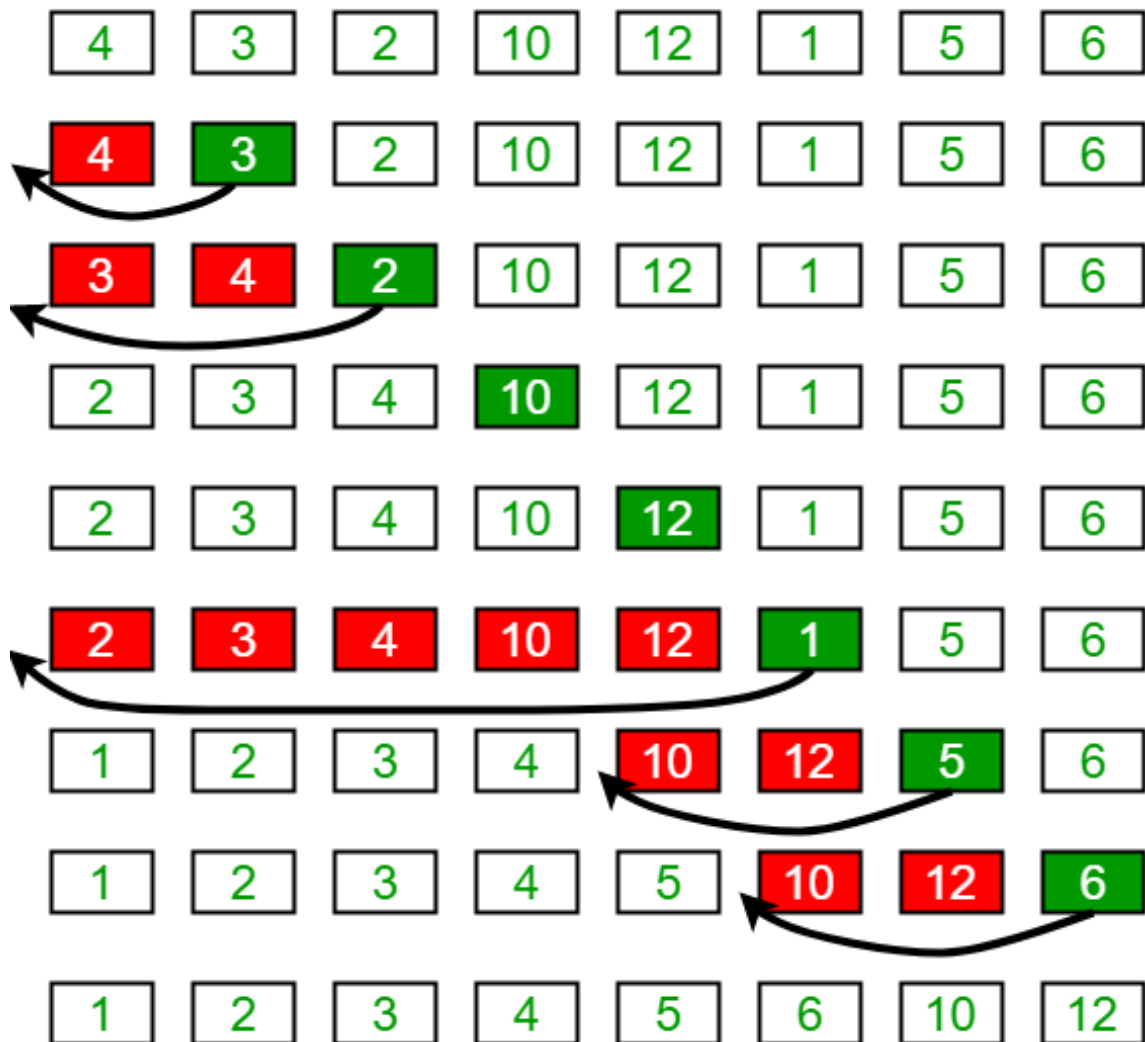
insertionSort(arr, n)

Loop from i = 1 to n-1.

.....a) Pick element arr[i] and insert it into sorted sequence arr[0...i-1]

### Example:

## Insertion Sort Execution Example



### Another Example:

12, 11, 13, 5, 6

Let us loop for  $i = 1$  (second element of the array) to 5 (Size of input array)

$i = 1$ . Since 11 is smaller than 12, move 12 and insert 11 before 12

11, 12, 13, 5, 6

$i = 2$ . 13 will remain at its position as all elements in  $A[0..i-1]$  are smaller than 13

11, 12, 13, 5, 6

$i = 3$ . 5 will move to the beginning and all other elements from 11 to 13 will move one position ahead of their current position.

5, 11, 12, 13, 6

$i = 4$ . 6 will move to position after 5, and elements from 11 to 13 will move one position ahead of their current position.

5, 6, 11, 12, 13

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

## C/C++

```
// C program for insertion sort
#include <stdio.h>
#include <math.h>

/* Function to sort an array using insertion sort*/
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i-1;

        /* Move elements of arr[0..i-1], that are
           greater than key, to one position ahead
           of their current position */
        while (j >= 0 && arr[j] > key)
        {
            arr[j+1] = arr[j];
            j = j-1;
        }
        arr[j+1] = key;
    }
}

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

/* Driver program to test insertion sort */
int main()
{
    int arr[] = {12, 11, 13, 5, 6};
    int n = sizeof(arr)/sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

[Run on IDE](#)

## Python

# Python program for implementation of Insertion Sort

```
# Function to do insertion sort
def insertionSort(arr):

    # Traverse through 1 to len(arr)
    for i in range(1, len(arr)):

        key = arr[i]

        # Move elements of arr[0..i-1], that are
        # greater than key, to one position ahead
        # of their current position
        j = i-1
        while j >=0 and key < arr[j] :
            arr[j+1] = arr[j]
            j -= 1
        arr[j+1] = key

# Driver code to test above
arr = [12, 11, 13, 5, 6]
insertionSort(arr)
print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i])

# This code is contributed by Mohit Kumra
```

[Run on IDE](#)

## Java

```
// Java program for implementation of Insertion Sort
class InsertionSort
{
    /*Function to sort array using insertion sort*/
    void sort(int arr[])
    {
        int n = arr.length;
        for (int i=1; i<n; ++i)
        {
            int key = arr[i];
            int j = i-1;

            /* Move elements of arr[0..i-1], that are
            greater than key, to one position ahead
            of their current position */
            while (j>=0 && arr[j] > key)
            {
                arr[j+1] = arr[j];
                j = j-1;
            }
            arr[j+1] = key;
        }
    }

    /* A utility function to print array of size n*/
    static 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
```

```
public static void main(String args[])
{
    int arr[] = {12, 11, 13, 5, 6};

    InsertionSort ob = new InsertionSort();
    ob.sort(arr);

    printArray(arr);
}
/* This code is contributed by Rajat Mishra. */
```

[Run on IDE](#)

Output:

```
5 6 11 12 13
```

**Time Complexity:**  $O(n^2)$

**Auxiliary Space:**  $O(1)$

**Boundary Cases:** Insertion sort takes maximum time to sort if elements are sorted in reverse order. And it takes minimum time (Order of  $n$ ) when elements are already sorted.

**Algorithmic Paradigm:** Incremental Approach

**Sorting In Place:** Yes

**Stable:** Yes

**Online:** Yes

**Uses:** Insertion sort is used when number of elements is small. It can also be useful when input array is almost sorted, only few elements are misplaced in complete big array.

### What is Binary Insertion Sort?

We can use binary search to reduce the number of comparisons in normal insertion sort. Binary Insertion Sort find use binary search to find the proper location to insert the selected item at each iteration. In normal insertion, sort it takes  $O(i)$  (at  $i$ th iteration) in worst case. we can reduce it to  $O(\log i)$  by using binary search. The algorithm as a whole still has a running worst case running time of  $O(n^2)$  because of the series of swaps required for each insertion. Refer [this](#) for implementation.

### How to implement Insertion Sort for Linked List?

Below is simple insertion sort algorithm for linked list.

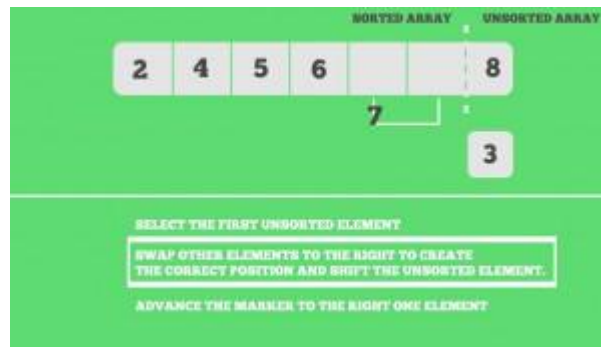
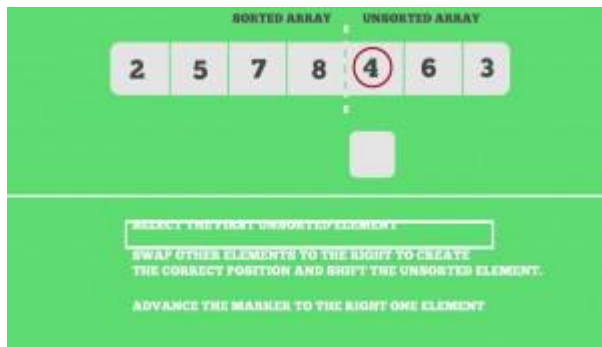
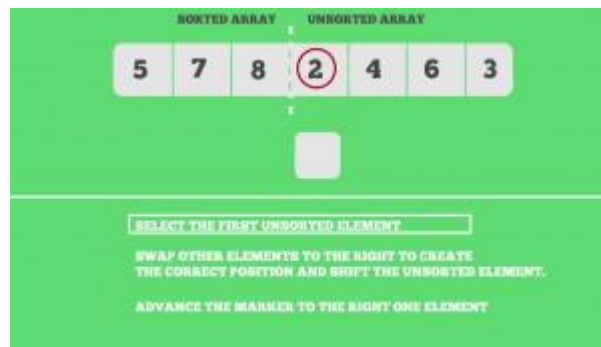
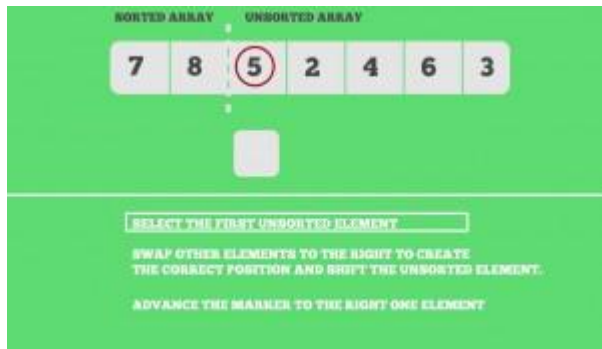
- 1) Create an empty sorted (or result) list
- 2) Traverse the given list, do following for every node.  
.....a) Insert current node in sorted way in sorted or result list.
- 3) Change head of given linked list to head of sorted (or result) list.

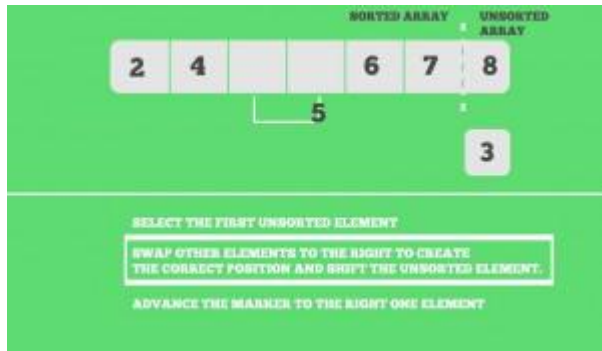
Refer [this](#) for implementation.

## Insertion Sort | GeeksforGeeks



### Snapshots:





## Quiz on Insertion Sort

### Other Sorting Algorithms on GeeksforGeeks/GeeksQuiz

Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Heap Sort, QuickSort, Radix Sort, Counting Sort, Bucket Sort, ShellSort, Comb Sort,

## Coding practice for sorting.

Image Source: [http://www.just.edu.jo/~basel/algorithms/Algo%20Slides/algo\\_ch2\\_getting\\_started.pdf](http://www.just.edu.jo/~basel/algorithms/Algo%20Slides/algo_ch2_getting_started.pdf)

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

## GATE CS Corner Company Wise Coding Practice

Sorting

Login to Improve this Article

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.

## Recommended Posts:

Merge Sort

Selection Sort

QuickSort

Bubble Sort

Recursive Insertion Sort

Pairs such that one is a power multiple of other

Loop Invariant Condition with Examples of Sorting Algorithms

Sort on the basis of number of factors using STL

Insertion sort to sort even and odd positioned elements in different orders

## Program for sorting variables of any data type

(Login to Rate)

2.1

Average Difficulty : 2.1/5.0  
Based on 66 vote(s)

☐

Add to TODO List

☐

Mark as DONE

Basic

Easy

Medium

Hard

Expert

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.


Share this post!

12 Comments

GeeksforGeeks

 Login

 Recommend

 Share

Sort by Newest



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



**DEVENDRA KASHYAP** • 19 days ago

some header files and variable are unnecessarily declared and defined in many programs..

^ | v • Reply • Share ›



**peter** • a month ago

Try use insert to middle sort, not need check full array.

Basic insert (worst variant)

12345 (5 elements)

(1) add 2, cmp 12

(12) add 3, cmp 13, 23

(123) add 4, cmp 14, 24, 34

(1234) add 5, cmp 15, 25, 35, 45

(12345)

Insert to middle (worst variant)

12345 (5 elements)

(1) add 2, mid=subarr.length/2=1/2=0.5, cmp 12

(12) add 3, mid=1, cmp 13, 23

(123) add 4, mid1=3/2=1.5, mid2=2/2=1, cmp 24, 34

(1234) add 5, mid1=4/2=2, mid2=3/2=1.5, cmp 35, 45, 15



(1234) add 5, mid1=4/2=2, mid2=2/2=1, cmp 35, 25, 15

(12345)

-2 cmp, but basic insert can be faster for some arrays

[see more](#)

^ | v • Reply • Share ›



**reenu saini** • 2 months ago

thanku

^ | v • Reply • Share ›



**Sachin Thakare** • 2 months ago

Python: I have a simpler logic, easy to understand and justifies the name 'insertion' sort.

L = [5,3,7,2,6,1,9]

=====

```
def sorting(L):
```

```
    ln = len(L)
```

```
    for i in range(1,ln):
```

```
        for j in range(i):
```

```
            if L[i] < L[j]:
```

```
                temp = L.pop(i)
```

```
                L.insert(j,temp)
```

```
            break
```

```
    return L
```

=====

Note : not sure why indents get removed in comments. How to avoid that, any advice ?

^ | v • Reply • Share ›



**amit** → Sachin Thakare • 2 months ago

please refer <http://304notmodified.com/D...>

^ | v • Reply • Share ›



**Ayaan Ali** • 3 months ago

Why including the math header file in c code?

^ | v • Reply • Share ›



**Fratyr** • 4 months ago

How come "while j <= 0 && key < arr[j]" ?

If key is the "next" to current position, then the last key will be nil, in Ruby you cannot compare nil to number. This algorithm is not generic.

^ | v • Reply • Share ›



**amit.jain@gmail.com** → Fratyr • 2 months ago

very well explaining in site <http://304notmodified.com/D...>

^ | v • Reply • Share ›



**Даниил Минаев** • 5 months ago

This is my example <https://github.com/minpor/sort>

^ | v • Reply • Share ›



**Mohit Makhija** • 6 months ago

For Source code in c++, python and java refer the link below

<http://code2begin.blogspot....>

^ | v • Reply • Share ›



**guest 2015** • 8 months ago

#include<stdio.h>

//Can we do like below..... I think it is the simplest code for insertion sort

```
int main()
{
    int a[]={19,12,10,8,6,4,1,0};
    int temp;
    int n=sizeof(a)/sizeof(a[0]);
    for(int i=1;i<n;i++) {="" for(int="" j="i-1;j">=0;j--)"
    {
        if(a[j]>a[i])
        {
            temp=a[j];
            a[i]=a[j];
            a[j]=temp;
            i--;
        }
    }

    for(int i=0;i<n;i++) {="" printf("%d\n",a[i]);="" }="" }="" >
1 ^ | v • Reply • Share ›
```



**Nishant Kumar** • 8 months ago

It should be i++ in outer loop (java)

for (int i=1; i<n; ++i)="">

^ | v • Reply • Share ›

## A computer science portal for geeks

710-B, Advant Navis Business Park,  
Sector-142, Noida, Uttar Pradesh - 201305  
[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

### COMPANY

About Us  
Careers  
Privacy Policy  
Contact Us

### PRACTICE

Company-wise  
Topic-wise  
Contests  
Subjective Questions

### LEARN

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

### CONTRIBUTE

Write an Article  
GBlog  
Videos

@geeksforgeeks, Some rights reserved









