

# GeeksQuiz

Computer science mock tests for geeks

## Insertion Sort

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



### 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:

**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**

```
// 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;
}
```

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.

**Quizzes:** [Sorting Questions](#)

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

Category: Searching and Sorting



Tweet

0

g+1

0

15 Comments

GeeksQuiz

1 Login ▾

♥ Recommend

🔗 Share

Sort by Best ▾



Join the discussion...



gwpark · a year ago

I love this sort rather than quick sort



I love this sort rather than quick sort.

In real world, an ARRAY or DATA to be sorted is usually NOT static, rather dynamic. If one element is inserted during sort process, other algorithms cannot respond easily. But, only this algorithm is not interrupted and can respond with the additional element.

23 ^ | v • Reply • Share ›



**Sumit Khatri** • a year ago

this is the sorting technique which can work even all the numbers to be sorted are not present in the memory at a time.....hence called online algorithm

11 ^ | v • Reply • Share ›



**Shubham Aggarwal** → Sumit Khatri • 4 months ago

I don't understand. Why is it called Online algorithm? Can u please link me to something deeper, when you said that numbers aren't in memory at the time it is running..

1 ^ | v • Reply • Share ›



**Sumit Khatri** → Shubham Aggarwal • 4 months ago

In all other algorithms you need all elements to be provided to the sorting algorithm before applying it. But an insertion sort allows you to start with partial set of elements, sorts it (called as partially sorted set), and if you want, you can insert more elements (these are the new set of elements that were not in memory when the sorting started) and sorts these elements too. I hope you got it. [http://en.wikipedia.org/wiki/Insertion\\_sort](http://en.wikipedia.org/wiki/Insertion_sort)

2 ^ | v • Reply • Share ›



**Haxor22** • 6 months ago

here a simple code is :

```
min=a[0];
```

```
for(i=1;i<n;i++) {for(j="0;j<i;j++") {if(a[i]<a[j]) {min="a[i];" a[i]="a[j];" a[j]="min;">
```

3 ^ | v • Reply • Share ›



**RB** → Haxor22 • 2 months ago

Thanx a lot!

^ | v • Reply • Share ›



**Logesh S** • 9 months ago

nice algorithm !!!

1 ^ | v • Reply • Share ›



**Torpedo** • 10 months ago

An IN-PLACE sorting algorithm is the one with following two properties:

- Input is usually overwritten by the output as the algorithm executes;
- Input is transformed using a Data Structure + small constant extra storage space for auxillary variables.

1 ^ | v • Reply • Share ›



**Torpedo** → Torpedo • 10 months ago

The auxillary space is given by GfG as  $O(1)$ .

^ | v • Reply • Share ›



**Goutham** • a year ago

here is a better understandable code for insertion sort :

```
for(i=0;i<n;i++){ n="" is="" the="" size="" of="" the="" array="" for(j="i;j">0;j--){
if(a[j-1]>a[j]){
temp=a[j-1];
a[j-1]=a[j];
a[j]=temp;
}
}
}
```

----- if i am wrong correct me :) :D

1 ^ | v • Reply • Share ›



**Aditi Rai** • 3 months ago

I want to know that whether my code comes under the category of insertion sort or not??

here is my code- please explain..

```
#include <stdio.h>
```

```
int main() {
```

```
int n;
```

```
scanf("%d",&n);
```

```
int a[n],i,j,t;
```

```
for (i = 0; i < n; i++)
```

```
scanf("%d",&a[i]);
```

```
for (i = 1; i < n; i++) {
```

```
for (j = i-1; j >= 0; j--) {
```

---

[see more](#)

^ | v • Reply • Share ›

**Ankit Bansal** • 6 months ago

```
public void insertion(int arg[]) {  
  
    int lenght = arg.length;  
  
    for (int i = 1; i < lenght; i++) {  
  
        int value = arg[i];  
  
        int j=i-1;  
  
        for (; j >= 0; j--) {  
  
            if (value < arg[j]) {  
  
                arg[j + 1] = arg[j];  
  
            } else {  
  
                break;  
  
            }  
  
        }  
  
        arg[j + 1] = value;  
  
    }  
  
}
```

^ | v • Reply • Share ›

**mast monsoon** • 8 months ago

is the above code not confusing??? i have difficulty in understanding that.... can we replace it by

```
for(i=0;i<n;i++) {="" for(j="i;j">=0;j--)  
{  
if(a(j)>a(j+1))  
  
swap a[j],a[j-1];  
}  
}
```

^ | v • Reply • Share ›

**Aditya** → mast monsoon • 7 months ago

Time complexity of your code will remain  $O(n^2)$ , irrespective of the input.

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Aman Mittal** • 3 months ago

```
public class InsertionSort {  
  
    private static void insertionSort(int arr[],int len) {  
  
        for (int i = 0; i < len; i++) {  
  
            for (int j = i+1; j < (len-1); j++) {  
  
                if (arr[i] > arr[j]) {  
  
                    arr[i] = arr[i] + arr[j];  
  
                    arr[j] = arr[i] - arr[j];  
  
                    arr[i] = arr[i] - arr[j];  
  
                }  
  
            }  
  
        }  
  
    }  
}
```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) › [Subscribe](#) [Add Disqus to your site](#) [Privacy](#)

Iconic One Theme | Powered by Wordpress