Insertion Sort Algorithm

Last Updated : 06 Aug, 2024

> ***Insertion sort*** *is a simple sorting algorithm that works by iteratively inserting each element of an unsorted list into its correct position in a sorted portion of the list. It is a* ***stable sorting*** *algorithm, meaning that elements with equal values maintain their relative order in the sorted output.*

Insertion sort is like sorting playing cards in your hands. You split the cards into two groups: the sorted cards and the unsorted cards. Then, you pick a card from the unsorted group and put it in the right place in the sorted group.

Insertion Sort Algorithm:

Insertion sort is a simple sorting algorithm that works by building a sorted array one element at a time. It is considered an ” **in-place** ” sorting algorithm, meaning it doesn’t require any additional memory space beyond the original array.

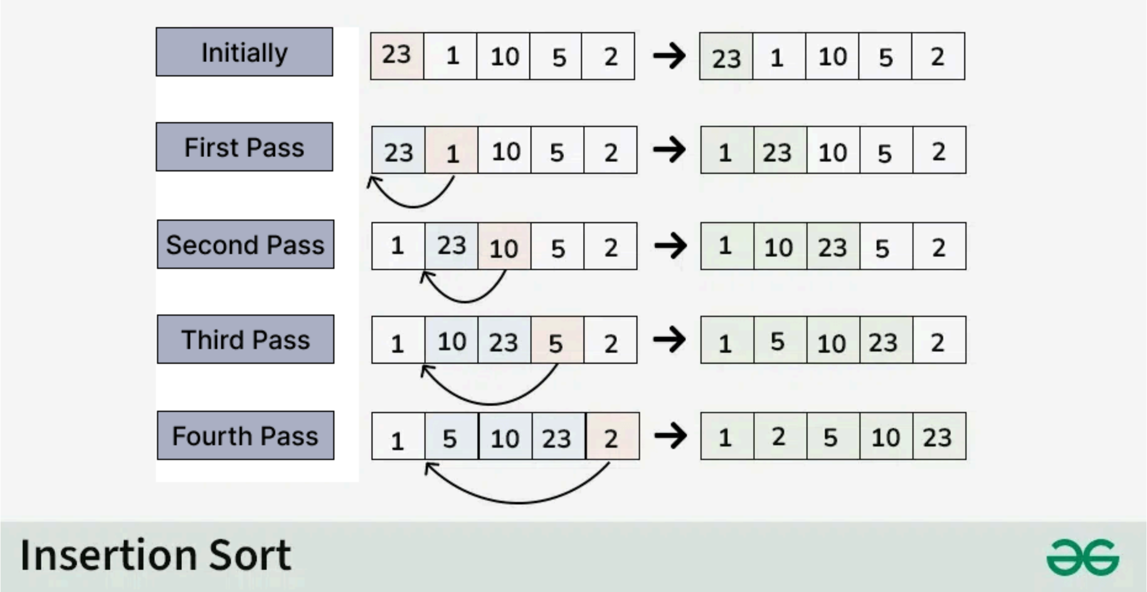
To achieve insertion sort, follow these steps:

- We start with second element of the array as first element in the array is assumed to be sorted.
- Compare second element with the first element and check if the second element is smaller then swap them.
- Move to the third element and compare it with the second element, then the first element and swap as necessary to put it in the correct position among the first three elements.

- Continue this process, comparing each element with the ones before it and swapping as needed to place it in the correct position among the sorted elements.
- Repeat until the entire array is sorted.

Working of Insertion Sort Algorithm:

Consider an array having elements ***: {23, 1, 10, 5, 2}***



Initial:

- *Current element is* ***23***
- *The first element in the array is assumed to be sorted.*
- *The sorted part until* ***0th*** *index is :* ***[23]***

First Pass:

- *Compare* ***1*** *with* ***23*** *(current element with the sorted part).*
- *Since* ***1*** *is smaller, insert* ***1*** *before* ***23*** *.*
- *The sorted part until* ***1st*** *index is:* ***[1, 23]***

Second Pass:

- *Compare **10** with **1** and **23** (current element with the sorted part).*
- *Since **10** is greater than **1** and smaller than **23** , insert **10** between **1** and **23** .*
- *The sorted part until **2nd** index is: **[1, 10, 23]***

Third Pass:

- *Compare **5** with **1** , **10** , and **23** (current element with the sorted part).*
- *Since **5** is greater than **1** and smaller than **10** , insert **5** between **1** and **10***
- *The sorted part until **3rd** index is **: [1, 5, 10, 23]***

Fourth Pass:

- *Compare **2** with **1, 5, 10** , and **23** (current element with the sorted part).*
- *Since **2** is greater than **1** and smaller than **5** insert **2** between **1** and **5** .*
- *The sorted part until **4th** index is: **[1, 2, 5, 10, 23]***

Final Array:

- *The sorted array is: **[1, 2, 5, 10, 23]***

Implementation of Insertion Sort:

C++ C Java Python C# JavaScript PHP

```
// C# program for implementation of Insertion Sort
using System;

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

        Console.WriteLine();
    }
```

DSA Course DSA Tutorial Data Structures Algorithms Array Strings Linked List Stack Que

```
        int[] arr = { 12, 11, 13, 5, 6 };

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

        printArray(arr);
    }
}

/* This code is contributed by Hritik Shah. */
```

Output

```
Number of passes: 4
5 6 11 12 13
```

Time Complexity: O(N^2)

Auxiliary Space: O(1)

Complexity Analysis of Insertion Sort :

Time Complexity of Insertion Sort

- **Best case: O(n)** , If the list is already sorted, where n is the number of elements in the list.
- **Average case: O(n ^{2})** , If the list is randomly ordered
- **Worst case: O(n ^{2})** , If the list is in reverse order

Space Complexity of Insertion Sort

- **Auxiliary Space:** O(1), Insertion sort requires **O(1)** additional space, making it a space-efficient sorting algorithm.

Advantages of Insertion Sort:

- Simple and easy to implement.
- Stable sorting algorithm.
- Efficient for small lists and nearly sorted lists.
- Space-efficient.
- Adoptive. the number of inversions is directly proportional to number of swaps. For example, no swapping happens for a sorted array and it takes O(n) time only.

Disadvantages of Insertion Sort:

- Inefficient for large lists.
- Not as efficient as other sorting algorithms (e.g., merge sort, quick sort) for most cases.

Applications of Insertion Sort:

Insertion sort is commonly used in situations where:

- The list is small or nearly sorted.
- Simplicity and stability are important.
- Used as a subroutine in Bucket Sort
- Can be useful when array is already almost sorted (very few inversions)

- Since Insertion sort is suitable for small sized arrays, it is used in Hybrid Sorting algorithms along with other efficient algorithms like Quick Sort and Merge Sort. When the subarray size becomes small, we switch to insertion sort in these recursive algorithms. For example IntroSort and TimSort use insertions sort.

Frequently Asked Questions on Insertion Sort

Q1. What are the Boundary Cases of the Insertion Sort algorithm?

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

Q2. What is the Algorithmic Paradigm of the Insertion Sort algorithm?

> *The Insertion Sort algorithm follows an incremental approach.*

Q3. Is Insertion Sort an in-place sorting algorithm?

> *Yes, insertion sort is an in-place sorting algorithm.*

Q4. Is Insertion Sort a stable algorithm?

> *Yes, insertion sort is a stable sorting algorithm.*

Q5. When is the Insertion Sort algorithm used?

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

GeeksforGeeks 785

Next Article

C Program For Insertion Sort

Similar Reads

Comparison among Bubble Sort, Selection Sort and Insertion Sort

Bubble Sort, Selection Sort, and Insertion Sort are simple sorting algorithms that are commonly used to sort small datasets or as building...

15 min read

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

We are given an array. We need to sort the even positioned elements in the ascending order and the odd positioned elements in the descending order....

7 min read

Sorting by combining Insertion Sort and Merge Sort algorithms

Insertion sort: The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct positio...

2 min read

Count swaps required to sort an array using Insertion Sort

Given an array A[] of size N (1 ≤ N ≤ 105), the task is to calculate the number of swaps required to sort the array using insertion sort algorithm....

15 min read

Merge Sort vs. Insertion Sort

Pre-requisite: Merge Sort, Insertion Sort Merge Sort: is an external algorithm based on divide and conquer strategy. In this sorting: The...

14 min read

Insertion Sort by Swapping Elements

Insertion Sort is suitable for arrays of small size. It also achieves the best-case complexity of O(n) if the arrays are already sorted. We have discusse...

11 min read

Insertion Sort Visualization using JavaScript

Insertion sort is a simple sorting algorithm in which values from the unsorted part are picked and placed at the correct position in the sorted...

5 min read

Sorting an Array in Bash using Insertion Sort

Given an array, arr[] of size N, the task is to sort the array in ascending order using Insertion Sort in bash scripting. Examples: Input: arr[] = {9, 7, ...

2 min read

Javascript Program For Insertion Sort In A Singly Linked List

We have discussed Insertion Sort for arrays. In this article we are going to discuss Insertion Sort for linked list. Below is a simple insertion sort...

3 min read

Binary Insertion Sort

Binary insertion sort is a sorting algorithm which is similar to the insertion sort, but instead of using linear search to find the location where an...

15+ min read

Article Tags : DSA Sorting Accenture Cisco +5 More

Practice Tags : Accenture Cisco Dell Grofers +4 More

GeeksforGeeks
Sanchhaya Education Private Limited
Corporate & Communications Address:- A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Inteview Preparation

Competitive Programming
Top DS or Algo for CP

UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects
Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos
DSA
Python
Java
C++
Web Development
Data Science
CS Subjects