Insertion Sort

Data Structures Made Easy

DUBLIN CITY UNIVERSITY

Contents

1. ***Definition***

Insertion sort is a comparison sorting algorithm that sorts one element at a time. Insertion sort takes a particular element from the array and places it in the correct index within the array. The process is continually executed until all items are correctly sorted in the array.

2. ***Implementation***

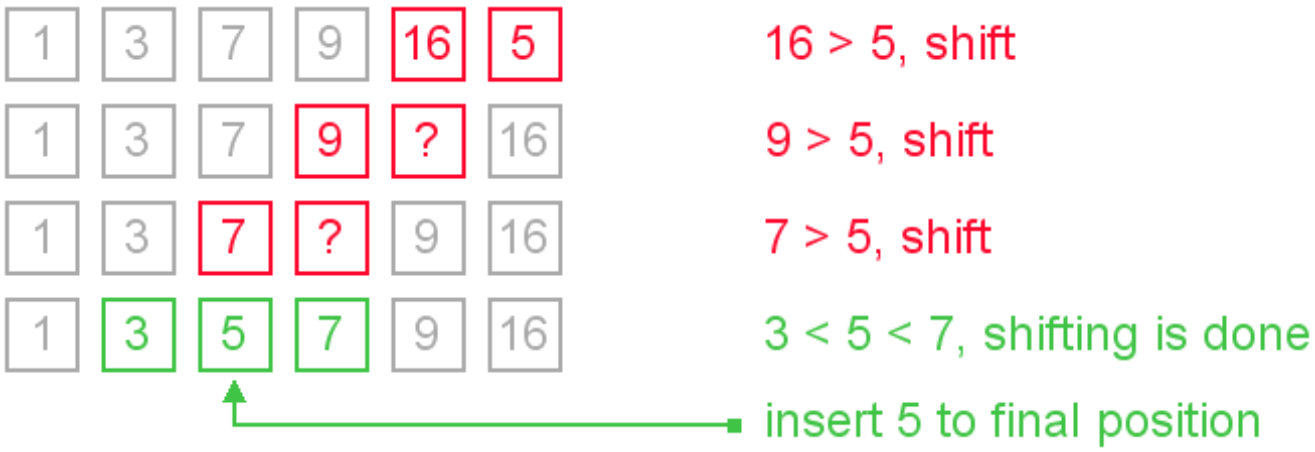
Insertion sort works as follows:

1. A selected element is sorted.

2. Pick the element next to the sorted sequence.
   - Move all elements in correctly sorted sequence (that are of a higher value than the element) one position right.
   - Place element within the gap.

3. Repeat above until all elements are sorted.

3. ***Example***



4. ***Functions***

The implementation of Insertion Sort is based on the following function

- **insertion ()**
  This function takes a selected element from the array and compares it with previous elements within the array. In order for elements to be exchanged, the element must be lesser than the previous element. That sort is then complete.

  The next element in the array is the selected element. The process repeats until the comparison has not reached the final index of the array.

5. ***Pseudocode***

```
function insertion ( array arr, size )

    for index 1 up to size of array
        element = arr [ index ]

        while index >  0 and element  <  arr [ index – 1 ]
            arr [ index ]  = arr [ index – 1]
            decrement index

        arr [ index ] = element
```

6. ***Complexity***

The optimal time complexity of Insertion Sort is O(N), which is only achieved when the array is already sorted. The amount of comparisons of elements within the array contributes to the worst time complexity of the algorithm, which is O(N ^ 2).

As insertion sort is an in-place algorithm, it therefore has a space-complexity of O(1).

7. ***Advantages of Insertion Sort***

Insertion sort is a sorting algorithm that shares similarities to bubble sort. However, insertion sort is more efficient because comparison of elements in insertion sort is less than the comparison of elements in bubble sort.

Insertion sort is an algorithm that is efficient for small data values and for data sets that are almost completely sorted.

Insertion sort is an in-place algorithm, which means it does not require additional memory space. This is effective when considering space complexity.

8. ***Disadvantages of Insertion Sort***

Insertion sort suffers from a computational complexity of O(N ^ 2). If the number of elements to be sorted increases, the performance of the application would be slow. As insertion sort requires a large number of shifts, the algorithm would be inefficient for applications that require the sorting of large amounts of data.

9. ***References***

Allison L. 2013. *Insertion Sort* [Online]. Available from: http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Sort/Insertion

Algorithms and Data Structures 2009. *Insertion Sort* [Online]. Available from: http://www.algolist.net/Algorithms/Sorting/Insertion_sort