



Insertion Sort Algorithm

Insertion sort works in the similar way as we sort cards in our hand in a card game.

We assume that the first card is already sorted then, we select an unsorted card.

If the unsorted card is greater than the card in hand, it is placed on the right otherwise, to the left.

In the same way, other unsorted cards are taken and put at their right place.

A similar approach is used by insertion sort.

Insertion sort is a sorting algorithm that places an unsorted element at its suitable place in each iteration.



How Insertion Sort Works?

Suppose we need to sort the following array.



1. The first element in the array is assumed to be sorted. Take the second element and store it separately in `key`.

Compare `key` with the first element. If the first element is greater than `key`, then `key` is placed in front of the first element.

step = 1



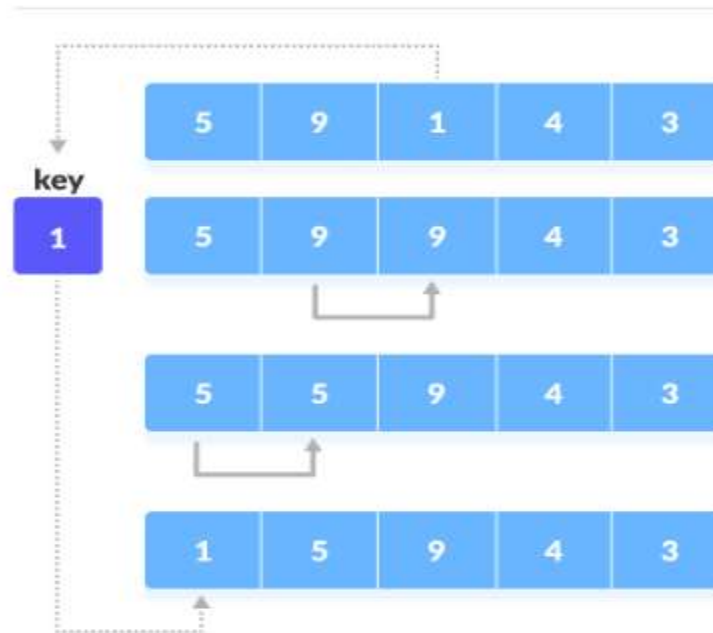


SNS COLLEGE OF TECHNOLOGY (AUTONOMOUS)



COIMBATORE - 35

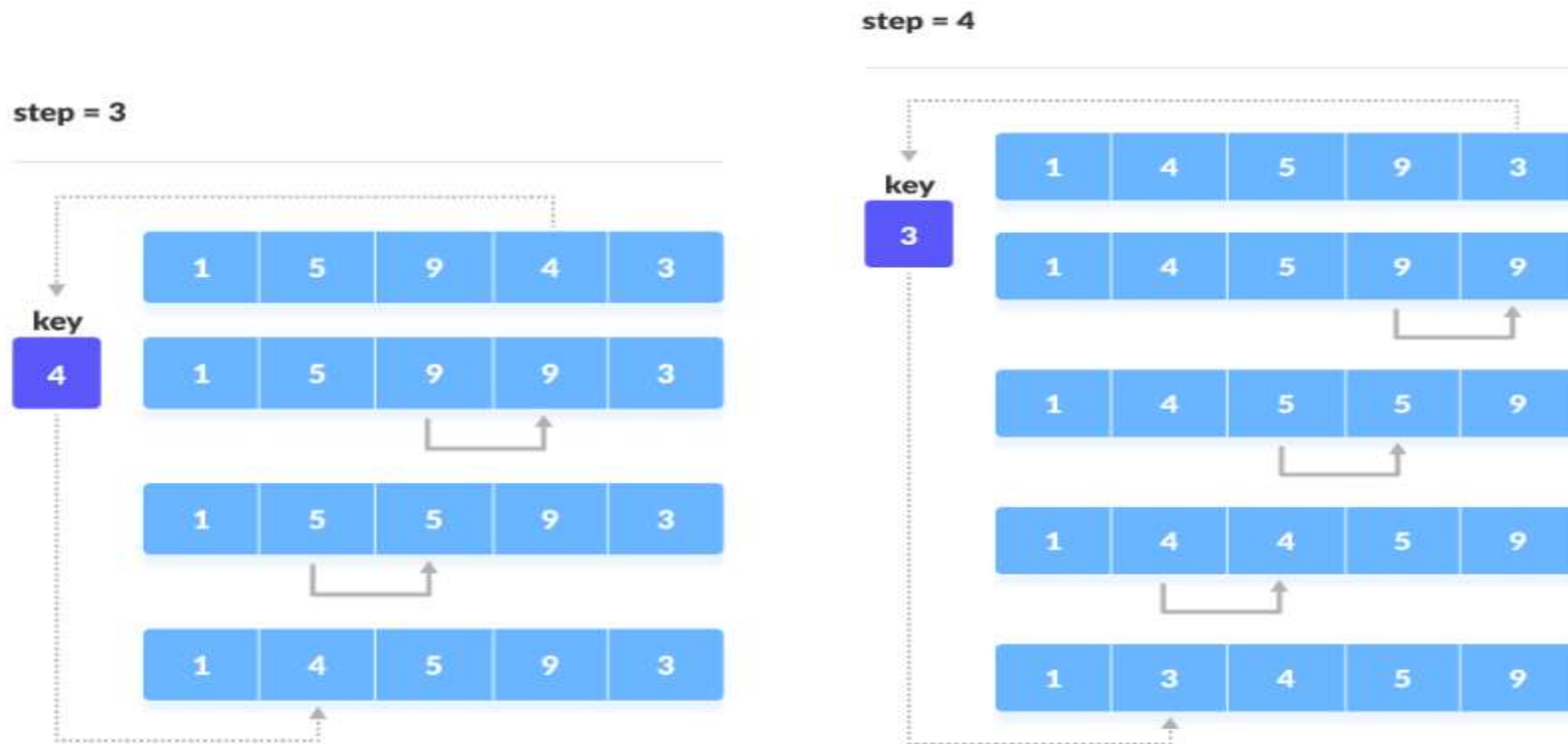
step = 2



3. In a similar way, place every unsorted element at its correct position.



3. In a similar way, place every unsorted element at its correct position.





Insertion Sort Algorithm

```
1. insertionSort(array)
2.   mark first element as sorted
3.   for each unsorted element X
4.     'extract' the element X
5.     for j <- lastSortedIndex down to 0
6.       if current element j > X
7.         move sorted element to the right by 1
8.     break loop and insert X here
9. end insertionSort
```



SNS COLLEGE OF TECHNOLOGY

(AUTONOMOUS)



COIMBATORE - 35

Time Complexities

•**Worst Case Complexity:** $O(n^2)$

Suppose, an array is in ascending order, and you want to sort it in descending order. In this case, worse case complexity occurs.

Each element has to be compared with each of the other elements so, for every n th element, $(n-1)$ number of comparisons are made.

Thus, the total number of comparisons = $n*(n-1) \sim n^2$

•**Best Case Complexity:** $O(n)$

When the array is already sorted, the outer loop runs for n number of times whereas the inner loop does not run at all. So, there is only n number of comparison. Thus, complexity is linear.

•**Average Case Complexity:** $O(n^2)$

It occurs when the elements of a array are in jumbled order (neither ascending nor descending).

Space Complexity

Space complexity is $O(1)$ because an extra variable key is used.



SNS COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

COIMBATORE - 35



Insertion Sort Applications

The insertion sort is used when:

- the array is has a small number of elements
- there are only a few elements left to be sorted