

Insertion Sort Algorithm

Insertion sort is [a sorting algorithm](#) that **places an unsorted element at its suitable place in each iteration.**

Insertion sort works similarly 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 in their right place.

A similar approach is used by insertion sort.

Working of Insertion Sort

Suppose we need to sort the following array.

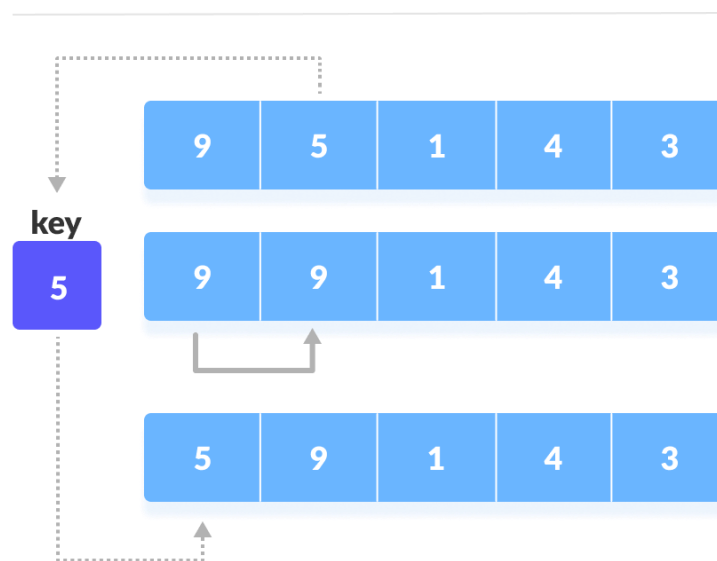
9	5	1	4	3
---	---	---	---	---

Initial 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.

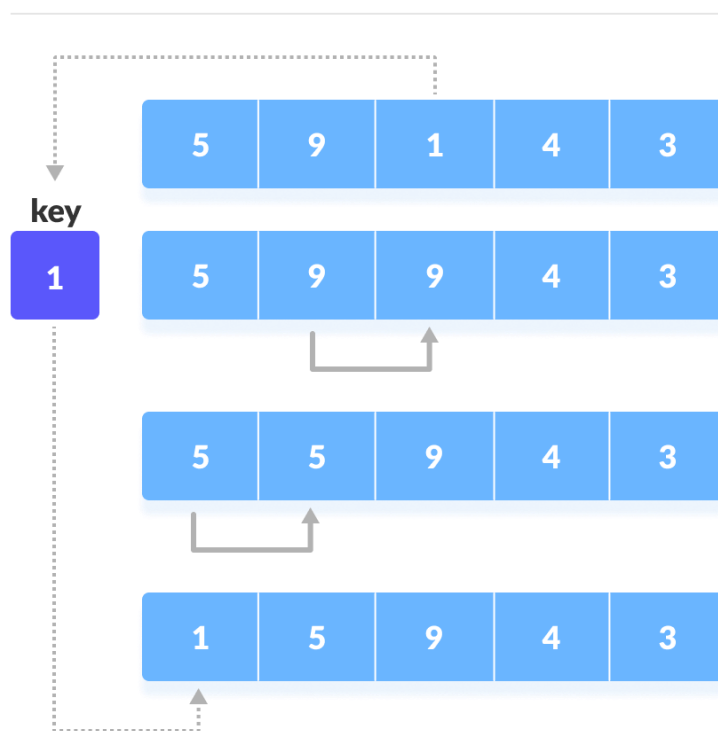
step = 1



2. Now, the first two elements are sorted.

Take the third element and compare it with the elements on the left of it. Placed it just behind the element smaller than it. If there is no element smaller than it, then place it at the beginning of the array.

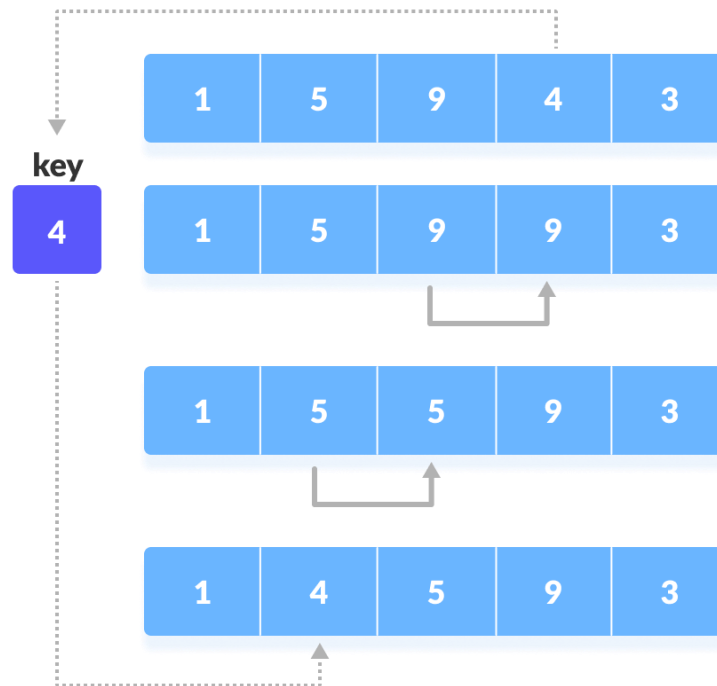
step = 2



Place 1 at the beginning

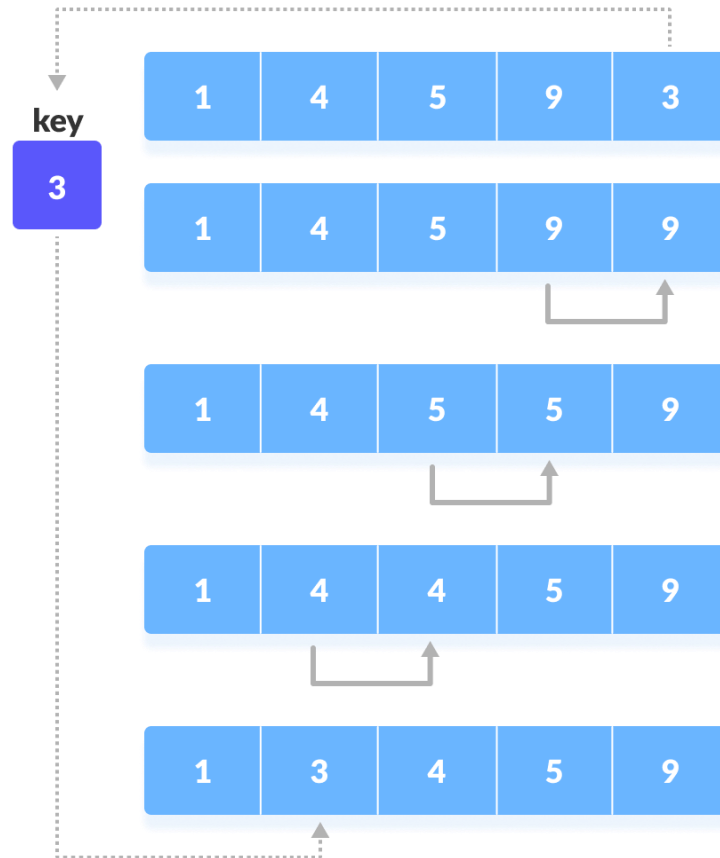
3. Similarly, place every unsorted element at its correct position.

step = 3



Place 4 behind 1

step = 4



Place 3 behind 1 and the array is sorted

Insertion Sort Algorithm

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