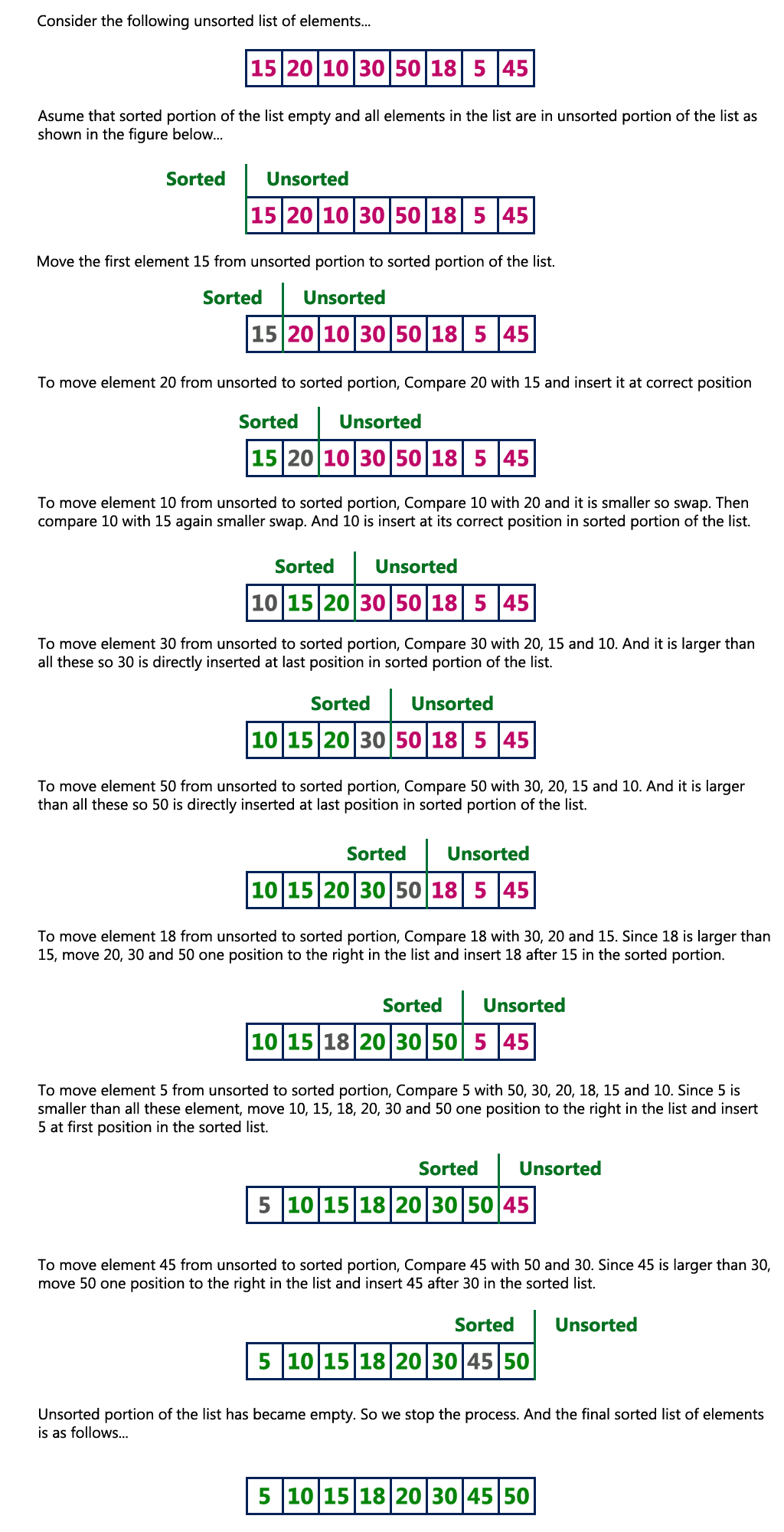
**Insertion Sort**

Insertion sort algorithm arranges a list of elements in a particular order. In insertion sort algorithm, every iteration moves an element from unsorted portion to sorted portion until all the elements are sorted in the list.



**Complexity of the Insertion Sort Algorithm**

**Worst Case : O(n2)**  
**Best Case : Ω(n)**  
**Average Case : Θ(n2)**

**Merge sort**

Merge sort is a sorting technique based on divide and conquer technique. Merge sort first divides the array into equal halves and then combines them in a sorted manner. Merge sort keeps on dividing the list into equal halves until it can no more be divided. By definition, if it is only one element in the list, it is sorted. Then, merge sort combines the smaller sorted lists keeping the new list sorted too.

Unsorted Array

We see here that an array of 8 items is divided into two arrays of size 4.

Merge Sort Division

 Now we divide these two arrays into halves.

Merge Sort Division

We further divide these arrays

Merge Sort Division

Now, we combine them in exactly the same manner as they were broken down.

We first compare the element for each list and then combine them into another list in a sorted manner. We see that 14 and 33 are in sorted positions. We compare 27 and 10 and in the target list of 2 values we put 10 first, followed by 27. We change the order of 19 and 35 whereas 42 and 44 are placed sequentially.

Merge Sort Combine

In the next iteration of the combining phase, we compare lists of two data values, and merge them into a list of found data values placing all in a sorted order.

Merge Sort Combine

After the final merging, the list should look like this −

Merge Sort

**Time complexity of Merge Sort is**nlog(n) in all three cases.