Insertion Sort

It works by iteratively inserting each element from the unsorted portion of the dataset into its correct position in the sorted portion of the dataset.

Time Complexity

best-case : O(N)

The dataset is already sorted.

worst-case : $O(N^2)$

The dataset is sorted but in the reverse order.

```
void insertionSort(int[] array)
	{
		for(int i=0;i<array.length-1;i++)
			{
				int j = i+1;
				while(j>0)
					{
						if(array[j]>=array[j-1])
							break;
						swap(array, j, j-1);
						j--;
					}
			}
	}
void swap(int[] array, int index1, int index2)
	{
		int temp = array[index1];
		array[index1] = array[index2];
		array[index2] = temp;
	}
```

Features

It is an *adaptive* algorithm. i.e.the steps are reduced when the dataset is sorted.

It is a *stable* algorithm.

It performs well when the dataset is partially sorted.