Name : Yash Srivastava || Section - A || Roll. No - 63

Implementation and Analysis of Merge Sort

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]

        merge_sort(L)
        merge_sort(R)
        i, j, k = 0, 0, 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1


print("Given array is: ")
a = list(map(int, input().split()))
merge_sort(a)
print("Sorted array is: ")
for i in range(len(a)):
    print(a[i], end=' ')
```

Name : Yash Srivastava || Section - A || Roll. No - 63

Input:

20 30 40 10 50 60

Output:

10 20 30 40 50 60

Time Complexity:

- O(nlogn) in all cases

Program:

Merge Sort.py

```
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]
        merge_sort(L)
        merge_sort(R)
        i, j, k = 0, 0, 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1
print("Given array is: ")
a = list(map(int, input().split()))
merge_sort(a)
print("Sorted array is: ")
for i in range(len(a)):
    print(a[i], end=' ')
```

Name : Yash Srivastava || Section - A || Roll. No - 63

Input/Output: