

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=' ')
```

Input:

20 30 40 10 50 60

Output:

10 20 30 40 50 60

Time Complexity:

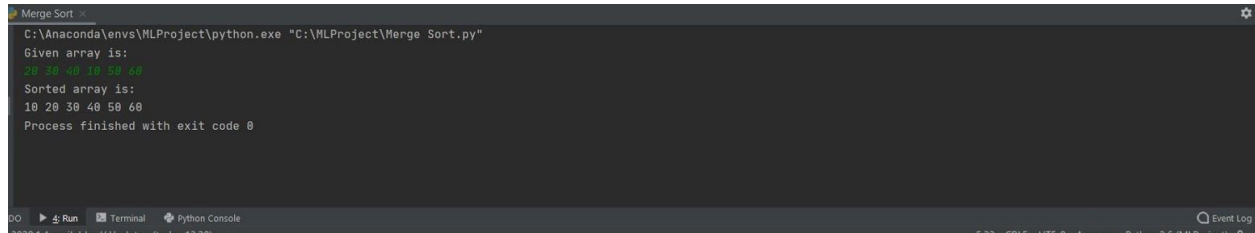
- $O(n \log n)$ in all cases

Program:

```
Merge Sort.py
1 def merge_sort(arr):
2     if len(arr) > 1:
3         mid = len(arr) // 2
4         L = arr[:mid]
5         R = arr[mid:]
6         merge_sort(L)
7         merge_sort(R)
8         i, j, k = 0, 0, 0
9         while i < len(L) and j < len(R):
10             if L[i] < R[j]:
11                 arr[k] = L[i]
12                 i += 1
13             else:
14                 arr[k] = R[j]
15                 j += 1
16             k += 1
17         while i < len(L):
18             arr[k] = L[i]
19             i += 1
20             k += 1
21         while j < len(R):
22             arr[k] = R[j]
23             j += 1
24             k += 1
25     print("Given array is: ")
26     a = list(map(int, input().split()))
27     merge_sort(a)
28     print("Sorted array is: ")
29     for i in range(len(a)):
30         print(a[i], end=' ')
```

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

Input/Output:



```
Merge Sort
C:\Anaconda\envs\MLProject\python.exe "C:\MLProject\Merge Sort.py"
Given array is:
20 30 40 10 50 60
Sorted array is:
10 20 30 40 50 60
Process finished with exit code 0
```

The screenshot shows a terminal window titled "Merge Sort". The command executed is `C:\Anaconda\envs\MLProject\python.exe "C:\MLProject\Merge Sort.py"`. The output displays the "Given array is:" followed by `20 30 40 10 50 60` (where 20, 30, and 40 are in green), then "Sorted array is:" followed by `10 20 30 40 50 60` (where 10, 20, and 30 are in green). The final line indicates "Process finished with exit code 0". The bottom status bar shows "Run", "Terminal", and "Python Console" tabs, along with a search icon and "Event Log".