

```
In [1]: import random
import time
import matplotlib.pyplot as plt
```

```
In [2]: def merge_sort(A, left, right):
    if left < right:
        mid = left + (right - left) // 2
        #print(A[left: right + 1], left, mid, right)
        merge_sort(A, left, mid)
        merge_sort(A, mid + 1, right)
        merge(A, left, mid, right)
```

```
In [9]: def merge(arr, l, m, r):
    n1 = m - l + 1
    n2 = r - m
    L = [0] * (n1)
    R = [0] * (n2)

    for i in range(0, n1):
        L[i] = arr[l + i]

    for j in range(0, n2):
        R[j] = arr[m + 1 + j]

    i = 0
    j = 0
    k = l

    while i < n1 and j < n2:
        if L[i] <= R[j]:
            arr[k] = L[i]
            i += 1
        else:
            arr[k] = R[j]
            j += 1
        k += 1

    while i < n1:
        arr[k] = L[i]
        i += 1
        k += 1

    while j < n2:
        arr[k] = R[j]
        j += 1
        k += 1
    #print(arr)
```

arr = [ 3, 1, 4 , 1, 2, 9, 0, 10] length = len(arr) n = length - 1 merge\_sort(arr, 0 , n)

```
In [10]: Best_case = [0, 1, 2, 3 , 4 , 5 , 6 , 7]
length = len(Best_case)
n = length - 1

st = time.time()
merge_sort(Best_case, 0 , n)
```

```
et = time.time()
elapsed_time = et - st
print('Best Case Execution time:', elapsed_time, 'seconds')
```

Best Case Execution time: 0.00020694732666015625 seconds

```
In [11]: worst_case = [0, 2, 4, 6, 1, 3, 5, 7]
length = len(worst_case)
n = length - 1

st = time.time()
merge_sort(worst_case, 0, n)
et = time.time()
elapsed_time = et - st
print('Worst Case Execution time:', elapsed_time, 'seconds')
```

Worst Case Execution time: 0.00020813941955566406 seconds

```
In [12]: avg_case = [0, 1, 2, 3, 7, 6, 5, 4]
length = len(avg_case)
n = length - 1

st = time.time()
merge_sort(avg_case, 0, n)
et = time.time()
elapsed_time = et - st
print('Avg Case Execution time:', elapsed_time, 'seconds')
```

Avg Case Execution time: 0.0002110004425048828 seconds

```
In [13]: input_list = [10, 1000, 5000, 10000]
time_taken = []

for i in input_list:
    random.seed(10)
    randomlist = random.sample(range(0, i), i)
    length = len(randomlist)
    n = length - 1
    st = time.time()
    merge_sort(randomlist, 0, n)
    et = time.time()
    elapsed_time = et - st
    time_taken.append(elapsed_time)
    print('When input is', i, ': Execution time:', elapsed_time, 'seconds')
```

When input is 10 : Execution time: 5.0067901611328125e-05 seconds  
 When input is 1000 : Execution time: 0.009556055068969727 seconds  
 When input is 5000 : Execution time: 0.025583982467651367 seconds  
 When input is 10000 : Execution time: 0.04241204261779785 seconds

```
plt.figure(figsize=(10,10)) plt.plot(input_list,time_taken, 'ro') plt.xticks(input_list)
plt.yticks(time_taken) plt.xlabel("input size") plt.ylabel("time taken for sorting")
plt.savefig('merge_sort.png') plt.show() plt.close()
```