```
from cProfile import label
import sys
import random
import math

from prettytable import PrettyTable
import matplotlib.pyplot as plt
import numpy as np

def initialize_counter():
   return [0]

def increment_counter(counter: list):
   counter[0] = counter[0] + 1

def generate_random_number(min: int, max: int):
   return random.randint(min, max)

def get_random_list(length: int):
   arr = []
   for i in range(length):
       arr.append(generate_random_number(0, 100))

   return arr

def merge(arr: list, begin: int, mid: int, end: int, counter: list):

   n1 = mid - begin + 1
   n2 = end - mid

   left_array = []
   for num in range(0, n1):
       left_array.append(arr[begin + num])

   right_array = []
   for num in range(0, n2):
       right_array.append(arr[mid + 1 + num])

   left_array.append(sys.maxsize)
   right_array.append(sys.maxsize)

   i = 0
```

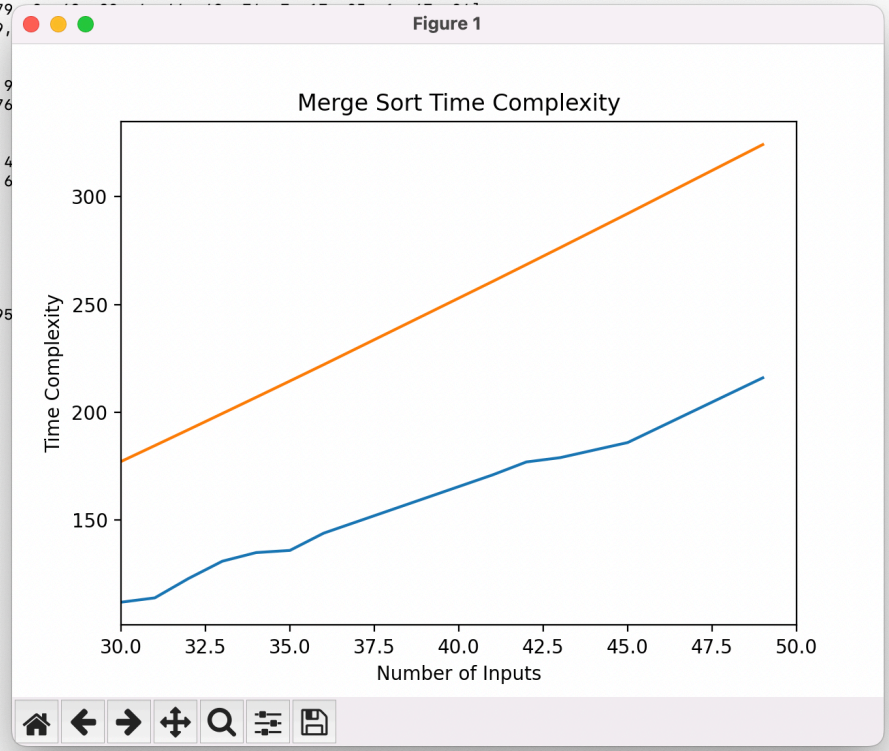
```
    j = 0
    k = begin


    while i < n1 and j < n2:
        if left_array[i] <= right_array[j]:
            increment_counter(counter)
            arr[k] = left_array[i]
            i += 1
        else:
            increment_counter(counter)
            arr[k] = right_array[j]
            j += 1
        k += 1

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

    while j < n2:
        arr[k] = right_array[j]
        j += 1
        k += 1



def merge_sort(arr: list, begin: int, end: int, counter: list):
    if begin < end:
        mid = begin + (end - begin) // 2

        merge_sort(arr, begin, mid, counter)

        merge_sort(arr, mid+1, end, counter)

        merge(arr, begin, mid, end, counter)


def __init__():
    # Minimum number of sets
    min_sets = 10
    # Maximum number of sets
```

```
    max_sets = 20

    number_of_sets = generate_random_number(min_sets, max_sets)

    table = PrettyTable(['Input', 'Actual Count', 'T(N)'])

    input_to_counter_dict = dict()

    for x in range(number_of_sets):
        # Initialize array size
        arr_size = generate_random_number(30, 50)

        # Initialize a random array
        arr = get_random_list(arr_size)

        # Print unsorted array
        print(f'Unsorted Array  : {arr}')

        counter = initialize_counter()

        # Sort the array
        merge_sort(arr, 0, arr_size - 1, counter)

        # Print sorted array
        print(f'Sorted Array   : {arr}')

        input_to_counter_dict[arr_size] = counter[0]

        table.add_row([arr_size, input_to_counter_dict[arr_size], round(arr_size +
(arr_size * math.log(arr_size, 2)), 2)])

        print('\n')

    print('\n\n')

    print(table)

    input_array = []
    counter_array = []
    avg_time_complexity_array = []

    for key in input_to_counter_dict:
```

```
        input_array.append(key)
        counter_array.append(input_to_counter_dict[key])
        avg_time_complexity_array.append(round(key + (key * math.log(key, 2)), 2))

    input_array.sort()
    counter_array.sort()
    avg_time_complexity_array.sort()


    x1 = np.array(input_array)
    y1 = np.array(counter_array)
    y2 = np.array(avg_time_complexity_array)


    plt.xlabel('Number of Inputs')
    plt.ylabel('Time Complexity')
    plt.title('Merge Sort Time Complexity')

    plt.xlim(30, 50)

    plt.plot(x1, y1, label='Actual Count')

    plt.plot(x1, y2, label='T(N)')

    print('\n\n')
    print(plt.show())

__init__()
```

assignment-2-merge-sort — Python merge-sort.py — 196×56

```
Sorted Array   : [1, 2, 3, 3, 4, 4, 9, 9, 10, 13, 14, 14, 17, 19, 19, 19, 19, 28, 31, 34, 38, 41, 41, 41, 43, 45, 47, 52, 53, 56, 61, 66, 67, 69, 74, 75, 75, 76, 77, 78, 78, 81, 82, 82, 83, 87, 93
, 94, 96]


Unsorted Array  : [14, 27, 73, 59, 24, 76, 58, 57, 63, 31, 88, 10, 100, 14, 58, 40, 85, 77, 88, 23, 86, 91, 84, 24, 22, 66, 25, 69, 4, 79, 88, 80, 18, 40, 12]
Sorted Array   : [4, 10, 12, 14, 14, 18, 22, 23, 24, 24, 25, 27, 31, 40, 40, 57, 58, 58, 59, 63, 66, 69, 73, 76, 77, 79, 80, 84, 85, 86, 88, 88, 88, 91, 100]


Unsorted Array  : [41, 16, 38, 93, 45, 52, 78, 40, 55, 42, 70, 40, 3, 80, 41, 57, 23, 42, 85, 27, 68, 68, 35, 23, 96, 5, 88, 34, 5, 44, 23, 57, 60, 46, 88, 25, 56, 54, 4, 93, 98, 28]
Sorted Array   : [3, 4, 5, 5, 16, 23, 23, 23, 25, 27, 28, 34, 35, 38, 40, 40, 41, 41, 42, 42, 44, 45, 46, 52, 54, 55, 56, 57, 57, 60, 68, 68, 70, 78, 80, 85, 88, 88, 93, 93, 96, 98]


Unsorted Array  : [45, 95, 37, 63, 38, 7, 71, 93, 7, 99, 69, 43, 57, 97, 81, 51, 77, 95, 79
Sorted Array   : [1, 2, 4, 7, 7, 7, 17, 25, 37, 38, 42, 43, 44, 45, 51, 57, 63, 67, 68, 69,


Unsorted Array  : [88, 35, 53, 85, 74, 79, 76, 39, 21, 15, 84, 29, 4, 53, 88, 81, 28, 40, 9
Sorted Array   : [4, 15, 15, 21, 24, 27, 28, 29, 35, 35, 38, 39, 40, 53, 53, 69, 74, 76, 76


Unsorted Array  : [65, 43, 87, 62, 99, 76, 88, 24, 29, 70, 47, 24, 90, 20, 96, 66, 7, 65, 4
Sorted Array   : [3, 4, 5, 7, 17, 19, 20, 24, 24, 29, 30, 36, 43, 44, 46, 47, 50, 54, 59, 6


Unsorted Array  : [19, 34, 43, 40, 86, 53, 38, 69, 69, 31, 88, 60, 52, 57, 51, 53, 80, 80,
Sorted Array   : [1, 1, 2, 17, 19, 23, 27, 29, 31, 31, 34, 37, 38, 39, 39, 40, 43, 47, 51,


Unsorted Array  : [28, 42, 77, 63, 18, 44, 68, 63, 61, 86, 93, 52, 5, 84, 79, 5, 97, 35, 95
Sorted Array   : [1, 5, 5, 14, 16, 18, 28, 35, 42, 44, 52, 59, 61, 63, 63, 63, 68, 72, 72,
```



Input	Actual Count	T(N)
45	186	292.13
30	112	177.21
41	171	260.66
43	177	276.33
34	135	206.97
33	126	199.47
31	116	184.58
49	216	324.12
35	136	214.52
42	179	268.48
32	123	192.0
31	113	184.58
36	144	222.12
33	131	199.47
31	114	184.58

```
/Users/rohitkrishnanvidyasagar/coding/university/Algo-analysis-assignments/assignment-2-merge-sort/merge-sort.py:140: MatplotlibDeprecationWarning: The resize_event function was deprecated in Matp
lotlib 3.6 and will be removed two minor releases later. Use callbacks.process('resize_event', ResizeEvent(...)) instead.
```