Leandro Lopez
Advanced Algorithms
Project 1

Merge Sort Algorithm

Merge sort is an **O(n log n)** class algorithm. While merge sort might not be the best performing algorithm in every single scenario, it should perform well in all scenarios, meaning that overall it has excellent performance. This means that, although a massive dataset will obviously take longer than a small dataset, we should expect the massive dataset to still perform well.

Array Size - 1000

```
        43321 function calls (41323 primitive calls) in 0.008 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        10.     0.000   0.000   0.008   0.008 <string>:1(<module>)
  1999/1        0.001   0.000   0.008   0.008 project1.py:35(merge_sort)
        999     0.005   0.000   0.007   0.000 project1.py:49(merge)
        1       0.000   0.000   0.008   0.008 {built-in method builtins.exec}
        22926   0.001   0.000   0.001   0.000 {built-in method builtins.len}
        8697    0.000   0.000   0.000   0.000 {method 'append' of 'list' objects}
        1       0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        8697    0.001   0.000   0.001   0.000 {method 'remove' of 'list' objects}
```

Array Size - 2000

```
        94558 function calls (90560 primitive calls) in 0.017 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1       0.000   0.000   0.017   0.017 <string>:1(<module>)
  3999/1        0.002   0.000   0.017   0.017 project1.py:35(merge_sort)
        1999    0.010   0.000   0.015   0.000 project1.py:49(merge)
        1       0.000   0.000   0.017   0.017 {built-in method builtins.exec}
        49769   0.002   0.000   0.002   0.000 {built-in method builtins.len}
        19394   0.001   0.000   0.001   0.000 {method 'append' of 'list' objects}
        1       0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        19394   0.001   0.000   0.001   0.000 {method 'remove' of 'list' objects}
```

Array Size - 3000

```
        148945 function calls (142947 primitive calls) in 0.028 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1        0.000   0.000   0.028   0.028 <string>:1(<module>)
  5999/1       0.003   0.000   0.028   0.028 project1.py:35(merge_sort)
        2999     0.017   0.000   0.025   0.000 project1.py:49(merge)
        1        0.000   0.000   0.028   0.028 {built-in method builtins.exec}
        78130 0.004   0.000   0.004   0.000 {built-in method builtins.len}
        30907 0.002   0.000   0.002   0.000 {method 'append' of 'list' objects}
        1        0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        30907 0.003   0.000   0.003   0.000 {method 'remove' of 'list' objects}
```

Array Size - 4000

```
        205080 function calls (197082 primitive calls) in 0.038 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1        0.000   0.000   0.038   0.038 <string>:1(<module>)
  7999/1       0.004   0.000   0.038   0.038 project1.py:35(merge_sort)
        3999     0.023   0.000   0.033   0.000 project1.py:49(merge)
        1        0.000   0.000   0.038   0.038 {built-in method builtins.exec}
  107525       0.005   0.000   0.005   0.000 {built-in method builtins.len}
        42777 0.002   0.000   0.002   0.000 {method 'append' of 'list' objects}
        1        0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        42777 0.003   0.000   0.003   0.000 {method 'remove' of 'list' objects}
```

Array Size - 5000

```
        263095 function calls (253097 primitive calls) in 0.049 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1        0.000   0.000   0.049   0.049 <string>:1(<module>)
  9999/1       0.005   0.000   0.049   0.049 project1.py:35(merge_sort)
        4999     0.030   0.000   0.043   0.000 project1.py:49(merge)
        1        0.000   0.000   0.049   0.049 {built-in method builtins.exec}
  137666       0.006   0.000   0.006   0.000 {built-in method builtins.len}
```

```
        55214  0.003   0.000   0.003   0.000 {method 'append' of 'list' objects}
        1         0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        55214  0.005   0.000   0.005   0.000 {method 'remove' of 'list' objects}
```

Array Size - 6000

```
        322074 function calls (310076 primitive calls) in 0.059 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1         0.000   0.000   0.059   0.059 <string>:1(<module>)
 11999/1        0.007   0.000   0.059   0.059 project1.py:35(merge_sort)
        5999    0.036   0.000   0.052   0.000 project1.py:49(merge)
        1         0.000   0.000   0.059   0.059 {built-in method builtins.exec}
  168365       0.008   0.000   0.008   0.000 {built-in method builtins.len}
        67854  0.003   0.000   0.003   0.000 {method 'append' of 'list' objects}
        1         0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        67854  0.006   0.000   0.006   0.000 {method 'remove' of 'list' objects}
```

Array Size - 7000

```
        381736 function calls (367738 primitive calls) in 0.070 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1         0.000   0.000   0.070   0.070 <string>:1(<module>)
 13999/1        0.008   0.000   0.070   0.070 project1.py:35(merge_sort)
        6999    0.042   0.000   0.062   0.000 project1.py:49(merge)
        1         0.000   0.000   0.070   0.070 {built-in method builtins.exec}
  199485       0.009   0.000   0.009   0.000 {built-in method builtins.len}
        80625  0.004   0.000   0.004   0.000 {method 'append' of 'list' objects}
        1         0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
        80625  0.007   0.000   0.007   0.000 {method 'remove' of 'list' objects}
```

Array Size - 8000

```
        442672 function calls (426674 primitive calls) in 0.082 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1         0.000   0.000   0.082   0.082 <string>:1(<module>)
```

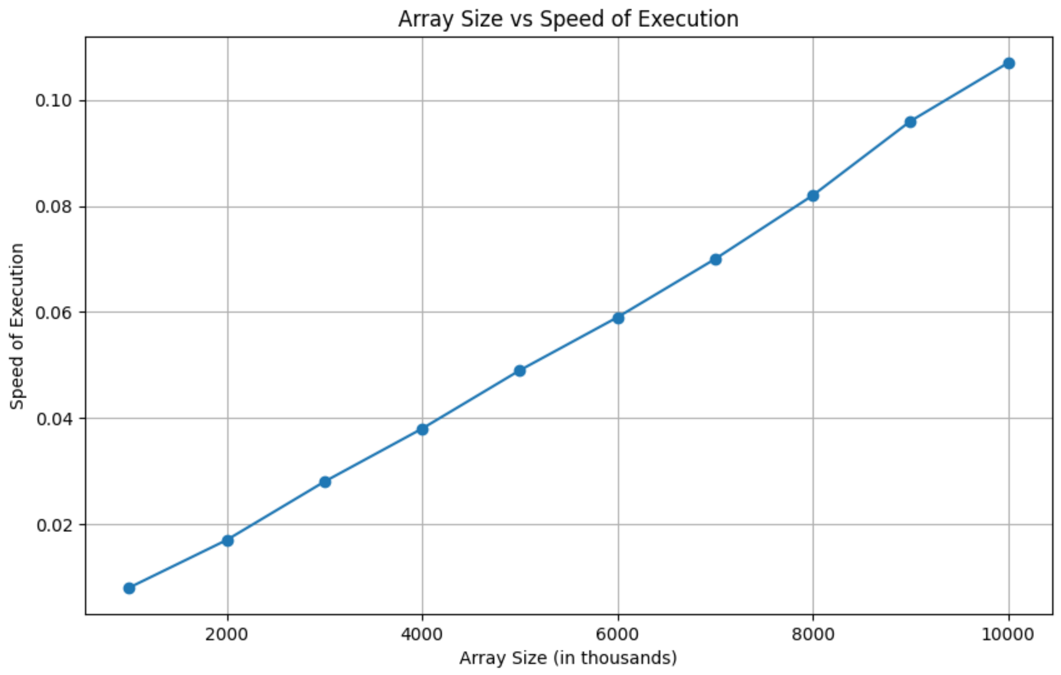
```
 15999/1      0.009   0.000   0.082   0.082 project1.py:35(merge_sort)
      7999    0.050   0.000   0.072   0.000 project1.py:49(merge)
      1       0.000   0.000   0.082   0.082 {built-in method builtins.exec}
  231345      0.011   0.000   0.011   0.000 {built-in method builtins.len}
      93663   0.005   0.000   0.005   0.000 {method 'append' of 'list' objects}
      1       0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
      93663   0.009   0.000   0.009   0.000 {method 'remove' of 'list' objects}
```

Array Size - 9000

```
      504116 function calls (486118 primitive calls) in 0.096 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1       0.000   0.000   0.096   0.096 <string>:1(<module>)
 17999/1      0.010   0.000   0.096   0.096 project1.py:35(merge_sort)
      8999    0.058   0.000   0.085   0.000 project1.py:49(merge)
      1       0.000   0.000   0.096   0.096 {built-in method builtins.exec}
  263169      0.012   0.000   0.012   0.000 {built-in method builtins.len}
  106973      0.005   0.000   0.005   0.000 {method 'append' of 'list' objects}
      1       0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
  106973      0.011   0.000   0.011   0.000 {method 'remove' of 'list' objects}
```

Array Size - 10000

```
      566536 function calls (546538 primitive calls) in 0.107 seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1       0.000   0.000   0.107   0.107 <string>:1(<module>)
 19999/1      0.011   0.000   0.107   0.107 project1.py:35(merge_sort)
      9999    0.064   0.000   0.094   0.000 project1.py:49(merge)
      1       0.000   0.000   0.107   0.107 {built-in method builtins.exec}
  295549      0.013   0.000   0.013   0.000 {built-in method builtins.len}
  120493      0.007   0.000   0.007   0.000 {method 'append' of 'list' objects}
      1       0.000   0.000   0.000   0.000 {method 'disable' of '_lsprof.Profiler' objects}
  120493      0.012   0.000   0.012   0.000 {method 'remove' of 'list' objects}
```



Conclusion

Our graph can help us visualize how Merge Sort works. We can see that the Speed of Execution increases in a linear fashion with the size of the Array. This means that, as we initially believed, our algorithm performs well. Our graph demonstrates that **O(n log n)** have a linear complexity.