Project 3 Report

Sahan Chery

Administrative

Group Name: Final Project Group 54

Member Name: Sahan Chery

Github Username: ceejaysc

Github Repository: https://github.com/ceejaysc/Project-3

Extended and Refined Proposal

Problem: Due to the fact that I had to do this project by myself, all I did was implement a merge sort and quick sort algorithm(s) that compared how fast each function performed in seconds.

Motivation: I was curious about the statistics of billionaires.

Features Implemented: Merge Sort, Quick Sort, and the comparison of how fast each algorithm runs.

Description of Data: The data is a multi-decade database of the super-rich. It builds off of the Forbes World's Billionaires list from 1996 to 2014. There are multiple variables for each billionaire like if they were self-made or inherited their wealth.

Tools/Languages/APIs/Libraries Used: Python, Pandas, Time Library, Billionaires data set and python code.

Algorithms Implemented: Quick Sort and Merge Sort

Distribution of Responsibility and Roles: Sahan Chery did everything

Analysis

- There has been many changes that doesn't follow the initial proposal. A person from my group dropped the class and the other didn't respond to my messages until recently. So I was forced to do this project by myself. Thankfully Professor Kapoor allowed me to implement and compare two algorithms from a dataset from Corgis to fulfill the project's requirements.
- For the merge_sort function the worst case time complexity is O(n log n) because it consistently divides the array into halves which is O(log n) and merges the sorted halves which is O(n), when combining them the worst case time complexity is O(n log n), where n is the number of elements in the dataset.
- For the quick_sort function, the time complexity in the worst case is $O(n^2)$ because the recursion tree can become n and at each level the partitioning process takes O(n), resulting in $O(n^2)$, where n is the number of elements in the dataset.

Reflection

- The overall experience for this project is was not great. I didn't have any help from my group since one dropped and the other doesn't respond to me when I message them. But being able to implement a quick and merge sort function was fun to say the least.
- I did have challenges when it comes to implementing the quick and merge sort functions. It took a bit of understanding and going back through the material to understand how to implement those two functions.
- If I were to start over as a group, I would like to have start the project earlier. Not only that I would have liked to be able to connect with my team a bit more.
- I learned how to implement two different algorithms/functions and compare them.

References

1. https://corgis-edu.github.io/corgis/python/billionaires/