Ayaan Bajwa
100864399
March 24, 2024

DSA Assignment 2:

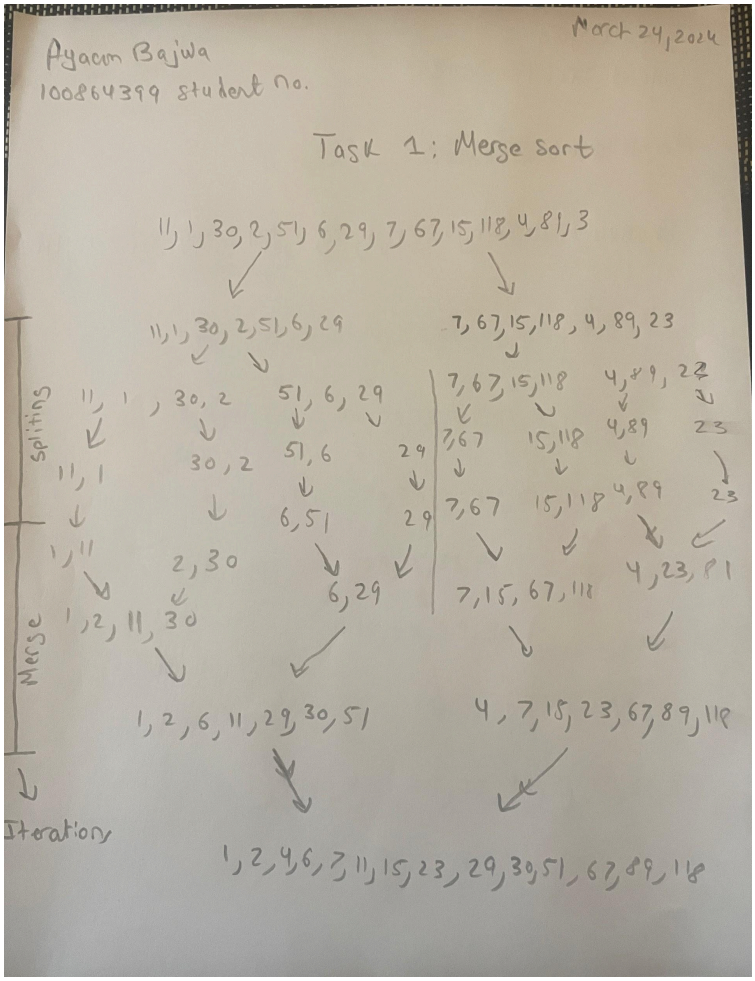
Sorting Algorithms Visualization and Simulation

Introduction/Objective:

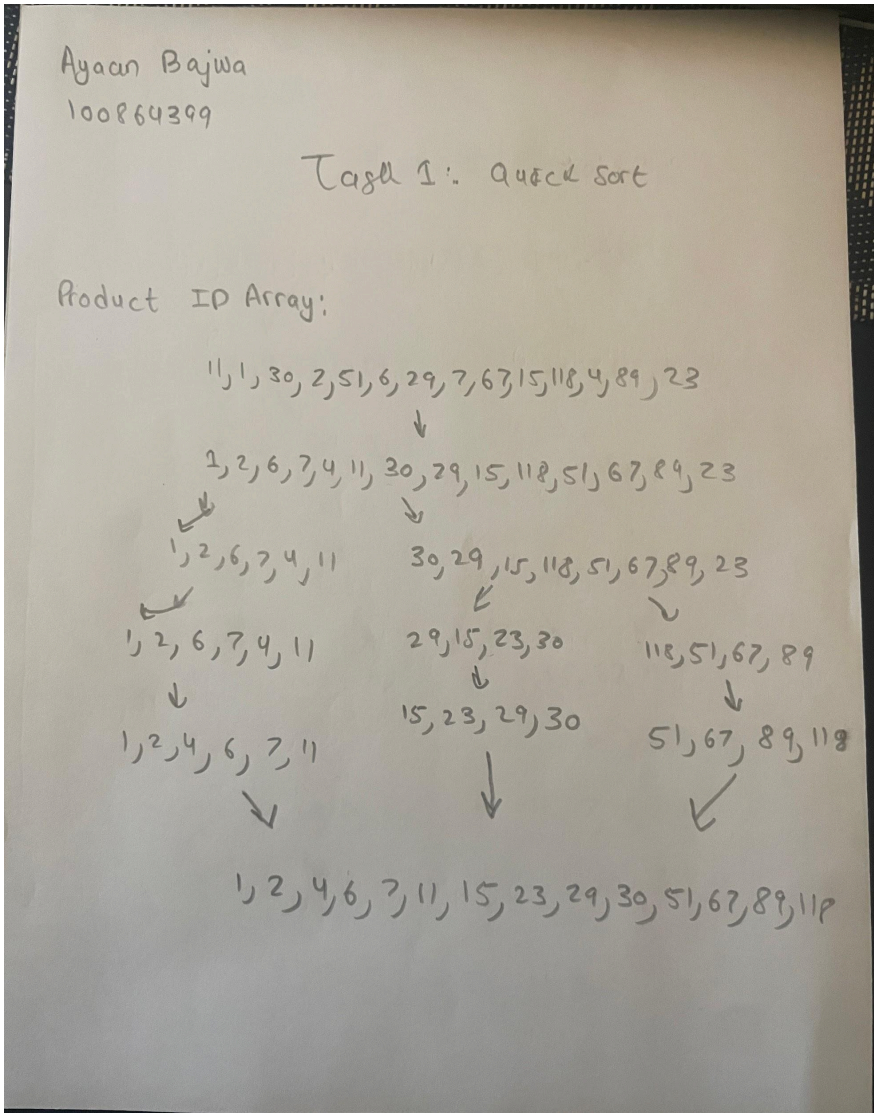
Sorting algorithms virtualization has a great influence over the computer science and software engineering roles and will gradually increase even more in the coming decade. Sorting algorithms play a vital role in arranging and organizing data of a certain type quickly and efficiently. Sorting is very fundamental for databases and search algorithms. Merge Sorting is one sorting algorithm that is commonly used and is a divide-and-conquer algorithm that recursively divides the unsorted list into smaller sublists. It will sort them independently each time and put them back together into one list to produce a sorted list. Merge Sort is known for its great time complexity, O(n log n). Also it is greatly known for its efficacy and reliability suiting it for large data sets such as online shopping platforms selling all sorts of items and making the user experience a memorable one. Quick Sort another algorithm that also follows the divide-and-conquer algorithm where the elements in the array are partitioned around it. Where elements in the array smaller go to the left and bigger to the right. The process is repeated and applied to the sub-arrays until the entire end of the array is completely sorted. This is also known for its great time complexity and efficiency. Quicksort can be greatly efficient to also use an online shopping experience by putting certain filters such as price, quality, or color and the quicksort algorithm goes through thousands and if not millions of products to find the preferred products and user is searching for. These algorithms greatly have a huge impact on our daily-lives to make it more convenient.

Detailed Visualization:

1. Merge Sort



2. Quick Sort



Conclusion:

After thorough visualization and implementation of sorting algorithms such as merge and quick sorting. There are many conclusions that can be drawn from. Through visualization and doing the process on the sheet above, it gave a deeper understanding of how both merge and quick sorting work and how different they can be but they all end up with the same answer and final product. With merge sort being more lengthy but easier to implement but quicksort being short but complicated. I learned both of them are very efficient with the time complexity of O(n log n). Some trade-offs are that the more complicated they are, the more memory space and CPU they use, and are very complex to implement. I learned that these sorting algorithms are used everywhere in

search engines, network optimization, and online retail stores to sort items by price, color, and quantity by quicksort. It is recommended for businesses to use the best sorting practices such as merge or quicksort as they will help the business immensely and create a better user experience, even though implementation complexity and costs may be a challenge.

Github Repository link:

https://github.com/aybaj75/Assign-2.git