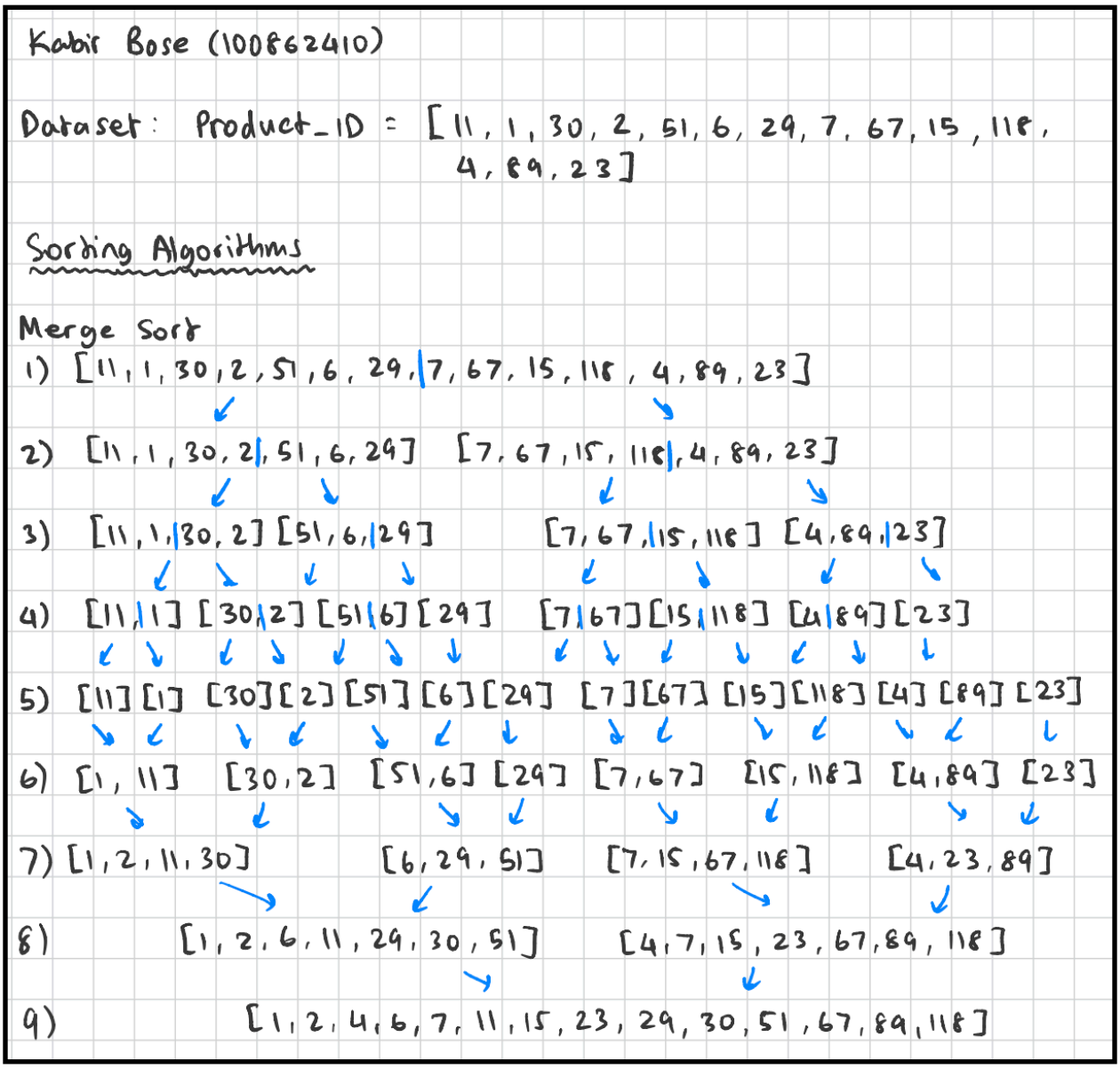
Objective

Sorting a given a dataset consisting of an array of product IDs with merge sort and quick sort, by showing every iteration for each algorithm.

Data Visualization

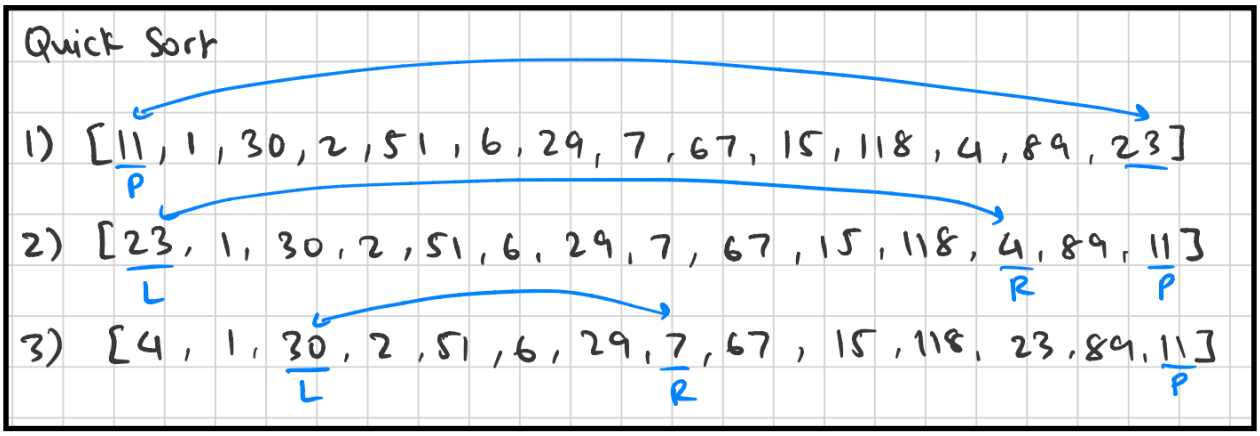
Merge sort

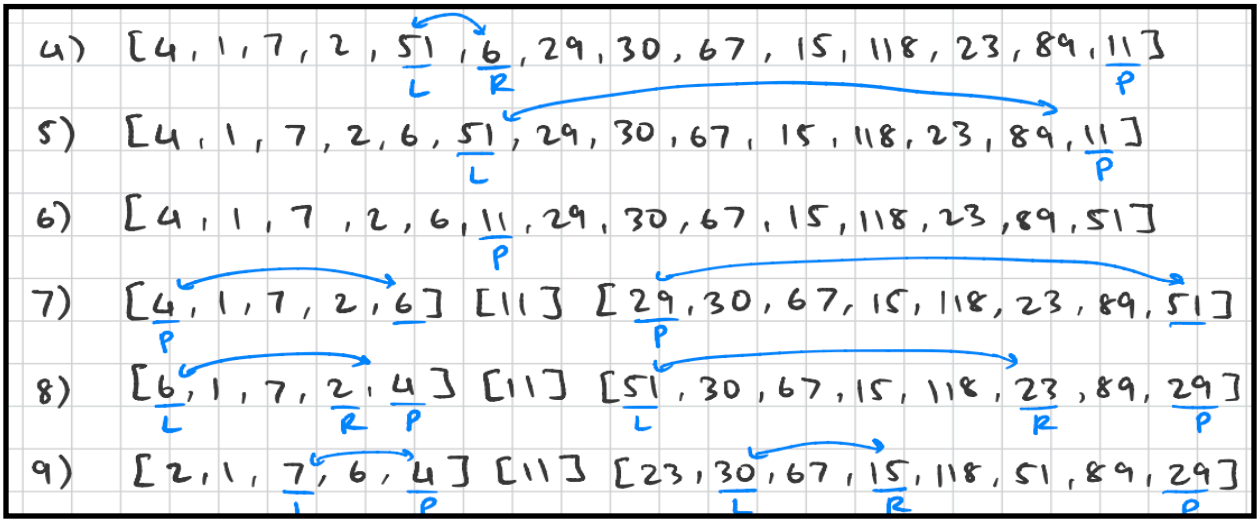
This is a demonstration of the recursive divide and conquer algorithm, merge sort. The below illustration demonstrates how merge sort works. Lines 1 to 5, demonstrate how merge sort divides an array of items into their individual items. Lines 6 to 9 demonstrate how merge sort merges items together while sorting them to create a sorted array.

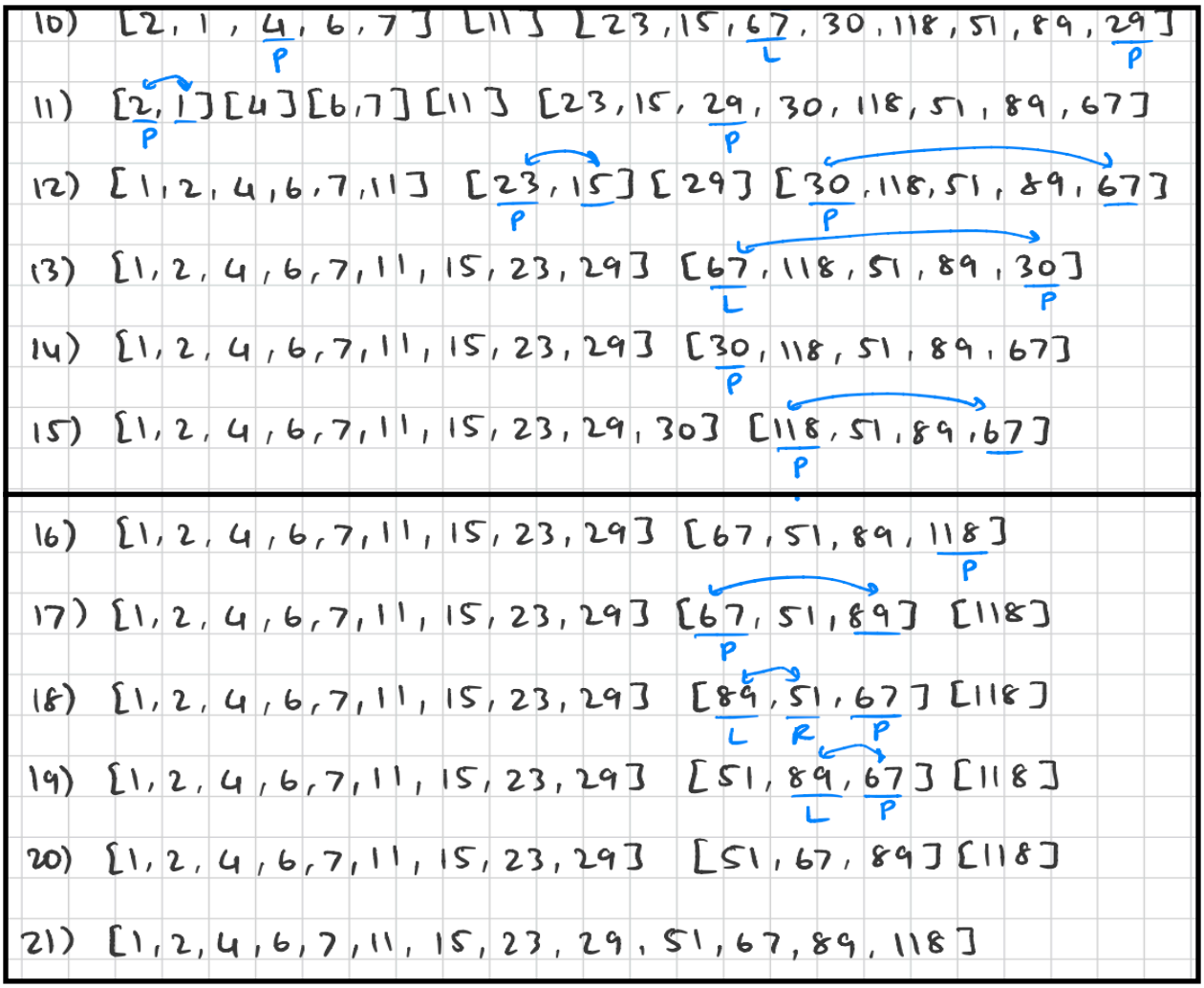


Quick Sort

This is a demonstration of the recursive divide and conquer algorithm, quick sort. The below illustration demonstrates how quick sort works. In line 1, we assign a pivot (denoted by *P*), in our case the first element in the array. In line 2, we must then swap the last element and our pivot, ensuring that the pivot is the last element. In line 3, we must find an element from the left that is greater than the pivot (denoted by *L*) and from the right that is less than the pivot (denoted by *R*). When there are no more *L* or *R*, we must swap *P* with *L*. These steps will be repeated till the array is sorted.







Time Complexity Comparison

Time Complexities	Merge Sort	Quick Sort
Best-case	O (n log n)	O (n log n)
Worst-case	O (n log n)	O (n^2)
Average-case	O (n log n)	O (n log n)

Code Implementation

For the full code, visit: https://github.com/KabirBose/merge-sort

```
# the og array
arr = [11,1,30,2,51,6,29,7,67,15,118,4,89,23]
```

```
kabirbose@Kabirs-MacBook-Air assignment-2 % python3 m
-----------------------------------------------
Array: [11, 1, 30, 2, 51, 6, 29, 7, 67, 15, 118, 4, 8
Left side: [11, 1, 30, 2, 51, 6, 29]
Right side: [7, 67, 15, 118, 4, 89, 23]
-----------------------------------------------
-----------------------------------------------
Array: [11, 1, 30, 2, 51, 6, 29]
Left side: [11, 1, 30]
Right side: [2, 51, 6, 29]
-----------------------------------------------
-----------------------------------------------
Array: [11, 1, 30]
Left side: [11]
Right side: [1, 30]
-----------------------------------------------
-----------------------------------------------
Array: [1, 30]
Left side: [1]
Right side: [30]
-----------------------------------
-----------------------------------
Array: [2, 51, 6, 29]
Left side: [2, 51]
Right side: [6, 29]
-----------------------------------
-----------------------------------
Array: [2, 51]
Left side: [2]
Right side: [51]
-----------------------------------
-----------------------------------
Array: [6, 29]
Left side: [6]
Right side: [29]
-----------------------------------
-----------------------------------
Array: [7, 67, 15, 118, 4, 89, 23]
Left side: [7, 67, 15]
Right side: [118, 4, 89, 23]
-----------------------------------
-----------------------------------------------
Array: [7, 67, 15]
Left side: [7]
Right side: [67, 15]
-----------------------------------------------
-----------------------------------------------
Array: [67, 15]
Left side: [67]
Right side: [15]
-----------------------------------------------
-----------------------------------------------
Array: [118, 4, 89, 23]
Left side: [118, 4]
Right side: [89, 23]
-----------------------------------------------
-----------------------------------------------
Array: [118, 4]
Left side: [118]
Right side: [4]
-----------------------------------------------
-----------------------------------------------
Array: [89, 23]
Left side: [89]
Right side: [23]
-----------------------------------------------

Old Result: [11, 1, 30, 2, 51, 6, 29, 7, 67, 15, 118, 4, 89, 23]
Final result: [1, 2, 4, 6, 7, 11, 15, 23, 29, 30, 51, 67, 89, 118]
```

Conclusion

From what I learned, merge sort seems to be a more efficient divide-and-conquer algorithm than quick sort. Merge sort has a more efficient worst-case time complexity (O (n log n)), while quick sorts' is (O (n^2)). Additionally, from the data visualization, merge sort seems much simpler than quick sort because there is no need for a pivot. With a larger dataset, quick sort would be very difficult to visualize compared to merge sort.