Jacob Lee

Problem 1 - Stability

Insertion Sort → A {1, 5, 2, 3, 0, 2, 2, 1, 4, 5}

1st iteration → {1, 5, 2, 3, 0, 2, 2, 1, 4, 5}

2nd iteration → {1, 2, 5, 3, 0, 2, 2, 1, 4, 5}

3rd iteration → {1, 2, 3, 5, 0, 2, 2, 1, 4, 5}

4th iteration → {0, 1, 2, 3, 5, 2, 2, 1, 4, 5}

5th iteration → {0, 1, 2, 2, 3, 5, 2, 1, 4, 5}

6th iteration → {0, 1, 2, 2, 2, 3, 5, 1, 4, 5}

7th iteration → {0, 1, 1, 2, 2, 2, 3, 5, 4, 5}

8th iteration → {0, 1, 1, 2, 2, 2, 3, 4, 5, 5}

9th iteration → {0, 1, 1, 2, 2, 2, 3, 4, 5, 5}

Merge Sort → A {1, 5, 2, 3, 0, 2, 2, 1, 4, 5}

Divide → {1, 5, 2, 3, 0} {2, 2, 1, 4, 5}

Conquer → {0, 1, 2, 3, 5} {1, 2, 2, 4, 5}

Merge → {0, 1, 1, 2, 2, 2, 3, 4, 5, 5}

Discussion

Both algorithms are stable as in the original array, there were multiple occurrences of {1}, {2}, and {5}, yet after sorting the order of all elements remained the same

Problem 2 - 3-way Merge sort

Merge Sort (A) ⟶ $T(n)$

C = Merge Sort $\left(A\left[1 : \frac{n}{3}\right]\right)$ ⟶ $T\left(\frac{n}{3}\right)$

D = Merge Sort $\left(A\left[\frac{n}{3}+1 : \frac{2n}{3}\right]\right)$ ⟶ $T\left(\frac{n}{3}\right)$

E = Merge Sort $\left(A\left[\frac{2n}{3}+1 : n\right]\right)$ ⟶ $T\left(\frac{n}{3}\right)$

B = Merge (C, D, E) ⟶ $f(n) = O(n)$

return B

$$T(n) = 3 \cdot T\left(\frac{n}{3}\right) + O(n)$$

$a = 3$, $b = 3$, $d = 1$

$a = 3 = b = 3^1$ = case 1

$O[n^1 \log n]$

= b) $O[n \log n]$