Time Complexity of 3-Way Mergesort

Introduction

To determine the order of growth of a **3-way mergesort**, we need to analyze how the algorithm behaves when dividing the array into three parts instead of two. The goal is to find the overall running time and compare it to the standard mergesort algorithm.

Step-by-Step Analysis

1. Dividing the Array

In a standard mergesort, we divide the array into two halves at each recursive step. For a 3-way mergesort, the array is divided into three equal parts. Each recursive call will sort one-third of the array, so instead of 2 recursive calls, we now make 3 recursive calls at each level.

2. Recurrence Relation

Let $T(n)$ represent the time complexity of sorting an array of size n. In the case of 3-way mergesort, the recurrence relation is:

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

The term $3T\left(\frac{n}{3}\right)$ comes from making 3 recursive calls, each on a subproblem of size $\frac{n}{3}$, and the $O(n)$ term represents the time required to merge the three sorted subarrays.

3. Solving the Recurrence

The recurrence relation can be solved using the **Master Theorem**. The recurrence is in the form:

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$$

where:

- $a = 3$ (number of subproblems),

- $b = 3$ (the size of each subproblem is $\frac{n}{3}$),
- $d = 1$ (merging takes linear time, $O(n)$).

To apply the Master Theorem, we compute the critical exponent:

$$\log_b a = \log_3 3 = 1$$

Now, compare d with $\log_b a$:

- If $d = \log_b a$, the solution to the recurrence is $O(n^d \log n)$.

Since $d = 1$ and $\log_3 3 = 1$, the time complexity is:

$$T(n) = O(n \log n)$$

Conclusion

The overall running time of 3-way mergesort is still $O(n \log n)$, similar to the standard 2-way mergesort. However, the base of the logarithm changes from 2 (in the case of standard mergesort) to 3, because the array is divided into three parts instead of two. The time complexity remains $O(n \log n)$, but technically, it is $O(n \log_3 n)$.

Since logarithms of different bases differ by a constant factor (i.e., $\log_3 n = \frac{\log_2 n}{\log_2 3}$), this constant factor does not affect the overall order of growth. Thus, the order of growth of the overall running time of the 3-way mergesort algorithm is:

$$O(n \log n)$$