

Assignment 2 – PS03 - Sugary Delight

Design approach:

An efficient recursive algorithm using Divide and Conquer is Merge Sort

Merge Sort is a popular and efficient sorting algorithm that utilizes the "divide and conquer" approach to sort a given array of elements. The idea behind Merge Sort is to divide the unsorted array into smaller sub-arrays, sort them, and then merge them back together to produce the sorted array. It is a recursive algorithm, which means that it breaks down the problem into smaller subproblems of the same type.

Here is a step-by-step explanation of how the Merge Sort algorithm works:

Step 1: Divide, The first step is to divide the unsorted array into two halves, and then recursively divide each half into smaller sub-arrays until each sub-array has only one element.

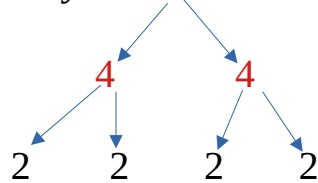
Step 2: Conquer, After dividing the array, we sort the individual sub-arrays by merging them. To merge two sub-arrays, we compare the first element of each sub-array and add the smaller element to a new array. We repeat this process until we have merged both sub-arrays into a single sorted array.

Step 3: Combine, The final step is to combine the sorted sub-arrays and produce the final sorted array. This is done by merging the sorted sub-arrays from the previous step.

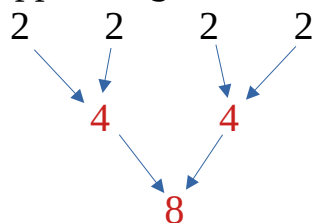
Time Complexity:

1. divide happens $\log N$ times

lets say $N = 8$



2. combine happens $\log N$ times



so for divide it takes $O(\log N)$

and combine also takes $O(\log N)$ but at each combine we merge the two sorted lists which takes $O(N)$ time so overall combine takes $O(N \log N)$

,continued..

finally we have

$$\begin{aligned}\text{Merge sort time} &= \text{time to divide} + \text{time to combine} \\ &= O(\log N) + O(N \log N) \\ &= O(\log N + N \log N) \\ &= O((1+N) \log N) \\ &= \sim O(N \log N)\end{aligned}$$

Alternate Solutions:

Divide and conquer: Quick sort with time complexity of $O(N \log N)$

Linear time: Radix Sort with time complexity of $O(N)$