

Name - Aashi Ashokbhai Goyani

ID - 1002205247

HOMEWORK - 4

PROBLEM - 1

2) Prove time complexity of the algorithm.

The 'Sorted-Array1' and 'Sorted-Array2' uses a divide and conquer strategy to merge sorted arrays.

- Divide and Conquer Recursion:-

It splits the input array in half at each level of recursion until there's only one element in each array. So, the total number of steps is determined by the logarithm of the number of input arrays ($O(\log K)$), where K is the number of arrays.

- Merging Sorted halves:-

The merge function combines two sorted

arrays of size N each into one sorted array. So, the time complexity of this operation is $O(N)$.

Therefore, the overall time complexity of the 'Sorted Array 1' and 'Sorted Array 2' function is $O(N * \log K)$

N is size of each array.

3) Ways to improve implementation

Memory Efficiency:-

The current implementation creates new arrays during the merge process, which could consume additional memory.

In-place merging techniques can be used to improve memory efficiency.

Iterative Merge:-

The recursive approach may lead to a large number of function calls, potentially causing a stack overflow for a large

number of input arrays. An iterative approach using a loop might be more memory-efficient and handle larger inputs.