

QUESTION 1

Problem 1 : Merge Intervals

Merge Intervals. An interval is a representation of a range of values along a number line. For example, $[1,3]$ represents all numbers between 1 and 3, inclusive. Given a collection of intervals, the task is to merge all overlapping intervals.

For example, consider the intervals $[1,3]$, $[2,6]$, $[8,10]$, $[15,18]$. The function should return merged intervals where overlapping intervals are combined. In this case, intervals $[1,3]$ and $[2,6]$ overlap, so they should be merged into $[1,6]$.

Short explanation of the approach:

1. Sort the intervals by their start times to process them in order.
2. Initialize a 'merged' list with the first interval.
3. For each interval:
 - Check for overlap with the last interval in 'merged':
 - If they overlap, merge them by updating the end time.
 - If not, add the current interval to 'merged'.
4. Return the 'merged' list as the final result.

This approach ensures overlapping intervals are combined efficiently, with a time complexity of **$O(n \log n)$** due to sorting.