

# School of Computer Science Engineering and Technology

<b>Course-B. Tech.</b>	<b>Type- Core</b>
<b>Course Code- CSE2307</b>	<b>Course Name- Design and Analysis of Algorithms</b>
<b>Year- 2025</b>	<b>Semester- Odd</b>
<b>Date- 10/09/2025</b>	<b>Batch- 2024-2028</b>

**Objective:** To understand and implement the Divide and Conquer technique in algorithms. To apply Divide and Conquer in solving search (Binary Search) and sorting (Merge Sort) problems. To analyze the complexity and efficiency of algorithms designed using Divide and Conquer.

**Theory:** Divide and Conquer is a problem-solving technique in computer science and mathematics. It breaks down a complex problem into smaller subproblems, solves each independently, and then combines the solutions to form the final answer. Steps involved:

- Divide – Break the given problem into smaller subproblems.
- Conquer – Solve the subproblems recursively. If the subproblem is small enough, solve it directly.
- Combine – Merge the solutions of the subproblems to get the overall solution.

**Q1.** Implement Binary search and determine the time required to search for an element. Write a program to perform Binary Search using Recursion on a sorted array of integers.

The program should:

- Accept elements in sorted order.
- Accept a key element to search.
- Use a recursive function to search.
- If the element is found, display its index. Otherwise, display "Element not found".

**Q2.** Implement maximum and minimum element search of an array and determine the time required to search for max and min elements. Write a program to perform min and max search using Recursion on a sorted array of integers.

The program should:

- Accept an array of integers in random order.
- Determine the time required for operations.

**Q3.** Write a program to sort a given set of elements using the Merge Sort algorithm and measure the time required for sorting. The program should:

- Accept or generate a list of integers of size n.
- Implement the Merge Sort algorithm to sort the list.
- Determine and record the execution time taken to sort the list.
- Repeat the experiment for different types of inputs a) Sorted List b) Unsorted List c) Random List and compare the execution time for each.
- Repeat the experiment for different sizes of n.
- Plot a graph of Time taken (y-axis) vs. Input size n (x-axis) to analyze the performance trend.