# **Assignment 2: Analyzing and Implementing Divide-and-Conquer Algorithms**

This assignment aims to deepen your understanding of divide-and-conquer algorithms, including their analysis and practical implementation. You will analyze the time complexity of algorithms using asymptotic notation and solve recurrence relations.

## Assignment Tasks:

1. **Asymptotic Analysis and Recurrence Relations (30 Points):**
   * Choose two divide-and-conquer algorithms, such as Quick Sort and Merge Sort. For each algorithm:
     + Describe the problem it solves and outline its key steps.
     + Provide a detailed analysis of its time complexity using O, Ω, and Θ notations.
     + Write the recurrence relation representing the algorithm's time complexity and solve it using the substitution method, recursion-tree method, and master method where applicable.
     + Discuss the practical implications of the algorithm's efficiency in real-world applications.
2. **Implementation and Comparison (20 Points):**
   * Implement the two chosen algorithms in a Python.
   * Compare the implementations by running them on various datasets, including sorted, reverse sorted, and random data.
   * Record and analyze the performance metrics, such as execution time and memory usage.
   * Discuss any observed discrepancies between theoretical analysis and practical performance, providing possible explanations.

## Formatting Guidelines:

* Your written analysis should be 3-4 pages in length, double-spaced, using a standard 12-point font (e.g., Times New Roman).
* Include a title page with your name, student ID, course title, and assignment number.
* Include well-commented code snippets and graphs/tables of performance metrics where relevant.
* Include a link to your GitHub repository.
* Cite any references used in APA format.

## Submission Instructions:

* Submit your written analysis and implementation code through the course's online portal.
* Ensure your analysis and explanations are clear, well-organized, and free of grammatical errors.