# Assignment on

# Analysis and Design of Algorithms

## 1. Divide & Conquer vs. Greedy vs. Dynamic Programming: A Comparative Analysis

- Analyze and compare the strengths and weaknesses of Divide & Conquer, Greedy, and Dynamic Programming (DP) for algorithm design.
- Consider the following factors:
    - **Problem types**: When is each approach best suited? What are the worst-case scenarios for each?
    - **Time & space complexity**: How do their complexities scale with problem size?
    - **Memory usage**: Do they require additional storage compared to basic approaches?
    - **Implementation complexity**: Are they easy to code and understand compared to other techniques?
- Provide specific examples of algorithms within each category to illustrate your points.

## 2. Design a Dynamic Programming (DP) Algorithm

Choose a specific problem and design an algorithm using DP to solve it.

- Clearly explain:
    - The problem definition, input, and desired output.
    - Your chosen DP approach and justify why it's suitable for this problem.
    - The subproblems you define and their relationships.
    - The recurrence relation you formulate to solve them.
    - The memoization strategy you employ to store solutions.
    - The overall algorithm flow and how it combines subproblem solutions.

**Bonus: Analyze the time and space complexity of your algorithm.**

**Submission Data: Nov, 8/2023**