**Department of Computer Science and Engineering**

**Chandpur Science and Technology University**

|  |  |
| --- | --- |
| Course Code: CSE 2201 | Credits: 1.50 |
| Course Name: Algorithm Design and Analysis | Semester: 2-2 |

**Lab 04**

**Greedy Algorithms – Coin Change Problem**

1. **Learning Objectives**

By the end of this lab, students should be able to:

* Understand the Greedy Algorithm paradigm.
* Implement the Coin Change algorithm using the greedy approach.
* Compare greedy solution with optimal solutions (if different).
* Analyze time and space complexity.

1. **Lesson Fit:**

Prerequisite: C/C++, Data Structure

1. **Theory Recap:**

### 

**Greedy Algorithms**

Greedy algorithms build up a solution piece by piece, always choosing the next piece that offers the most immediate benefit.

The key is to make a locally optimal choice at each step with the hope of finding the global optimum.

**Coin Change Problem**

Problem: Given a set of coin denominations and a target amount, find the minimum number of coins needed to make that amount.

**Greedy Strategy**: Always choose the largest coin value that is less than or equal to the remaining amount.

**Example:**

Denominations: [1, 5, 10, 25]

Target amount: 30

Greedy choice: 25 + 5 = 30 (2 coins)

When Greedy Fails

Greedy algorithms do not always provide the optimal solution.

**Example:** Denominations [1, 3, 4], Target: 6. Greedy gives 4+1+1 = 3 coins; optimal is 3+3 = 2 coins.

**Lab 4 Activity List**

# Experiment # 1: *Implement the Coin Change problem using a greedy approach and evaluate its correctness.*

**Tasks:**

* Write pseudocode for the greedy coin change algorithm.
* Implement the algorithm in C/C++/Python.
* Test on various denominations and amounts.
* Compare the result with the known optimal solution (where applicable).

**📊 Time Complexity:**

Time Complexity (Theoretical): O(n), where n is the number of denominations (assuming sorted input).

**🧮 Space Complexity:**

Space Complexity: O(1), constant extra space (or O(n) if storing coins used).

### 📊 Comparison Table (Empirical)

|  |  |  |  |
| --- | --- | --- | --- |
| Amount | Coins Used (Greedy) | Minimum Coins (Optimal) | Greedy Correct? |
|  |  |  |  |
|  |  |  |  |

***Report:***

The report should cover the following

Name of the Experiment

1. Objective
2. Algorithm
3. Theoretical Solution of given problem
4. Practical Work:
   1. Pseudocode
   2. Source Code in C/CPP/Python
5. Analysis Table

| Algorithm | Best Case | Worst Case | Avg Case | Space |

|----------------|-----------|------------|----------|--------|

1. Observations
2. Challenges
3. Conclusion

📸 Attachments:

- Screenshot of program output.

- Manual step count snapshots.

- Complexity graph (drawn or plotted).