CSE-2212: Design and Analysis of Algorithms-I Lab Practice Lab 10– January 13, 2025

Survey on DP : https://forms.gle/J7MNNC55MX9RJsUc7

Experiment: In this lab experiment, you will have to implement four algorithms in Java—

1) Rod Cutting (**Lecture 18 - Rod Cutting and Rock Climbing.pdf**)
2) Sum of Subset - https://www.shafaetsplanet.com/?p=3662
3) Coin Change - https://www.shafaetsplanet.com/?p=3638
4) Matrix Chain Multiplication (Lecture 19 - Matrix Multiplication.pptx.pdf), **https://www.shafaetsplanet.com/?p=3678**

Problem 1: Rod Cutting

You will implement a solution to determine the maximum profit obtainable by cutting a rod of length into smaller lengths, given the prices of each rod length.

Method to Implement:
maxProfit(int[] prices, int n): Returns the maximum profit obtainable for a rod of length n.
Example Input:
prices = [1, 5, 8, 9, 10, 17, 17, 20]
n = 8
Expected Output:
22 (Cut the rod into lengths 2 and 6 to get prices 5 + 17 = 22.)

Problem 2: Sum of Subset

You will determine whether there exists a subset of the given set that adds up to a specific target sum.

Method to Implement:
isSubsetSum(int[] set, int n, int target): Returns true if a subset exists with the target sum, otherwise false.
Example Input:
set = [3, 34, 4, 12, 5, 2]
target = 9
Expected Output:
true (Subset {4, 5} adds up to 9.)

Hint:
Build a table where the value at dp[i][j] indicates whether a sum of can be achieved with the first elements of the set.

Problem 3: Coin Change

You will find the number of ways to make change for a given amount using a set of coin denominations.

Method to Implement:

coinChangeWays(int[] coins, int n, int amount): Returns the number of ways to make change for the amount.

Example Input:

coins = [1, 2, 3]

amount = 4

Expected Output:

4 (Ways: {1,1,1,1}, {1,1,2}, {2,2}, {1,3})

Explanation:

Build a table where dp[i][j] represents the number of ways to make an amount using the first coins.

Problem 4: Matrix Chain Multiplication

You will find the minimum number of scalar multiplications needed to multiply a chain of matrices.

Method to Implement:

matrixChainOrder(int[] dims): Returns the minimum number of scalar multiplications needed.

Example Input:

dims = [1, 2, 3, 4]

Expected Output:

18 (Optimal parenthesization is ((A1A2)A3).)

Explanation:

Use dynamic programming to determine the minimum cost of multiplying matrices from index to in the chain. The value at dp[i][j] represents this cost.