

**Lab Exercise 7**

**Problem 1: Weighted Interval Scheduling.**

You will implement the Weighted Interval Scheduling problem discussed in class. The input is a set of  $n$  intervals: each interval has a start time, an end time, and an associated weight. The goal is to pick a subset of non-overlapping intervals whose total weight is maximal.

- a) Implement a memoized Top-Down (recursive) algorithm that takes as input ' $n$ ' intervals and prints the weight of the optimal solution.
- b) Write the memoized Bottom-Up (non-recursive) version of the previous algorithm.
- c) Print the selected intervals of the optimal solution.

**Problem 2: Coin-Changing.**

We saw in the previous lab that a greedy approach to the coin-change problem does not work in all cases. Use dynamic programming to solve the problem for the general case: For an input set of coins  $\{c_1, c_2, \dots, c_n\}$  find the minimal number of coins needed to make change for an amount  $S$ .

**Problem 3: Masoning Options.**

You are a mason working with bricks of four sizes (length x breadth x height) given as  $(1 \times 1 \times 1)$ ,  $(1 \times 1 \times 2)$ ,  $(1 \times 1 \times 3)$ , and  $(1 \times 1 \times 4)$ . There is an infinite number of bricks for each type at your disposal. You are asked to build a fence wall of width 1, whose height is ' $H$ ' and length is ' $L$ '. The wall should be structurally stable: it should not have any holes in it and you should not be able to cut it vertically (from top to bottom) without cutting one or more bricks.

Assuming that the blocks can only be placed horizontally, in how many ways can you build your wall? Your program should take as input values for ' $H$ ' and ' $L$ '.