SIT221 –DATA STRUCTURES AND ALGORITHMS

LAB11: DYNAMIC PROGRAMMING

LAB OBJECTIVE:

The objective of this lab is to develop your skills in solving problems using Dynamic Programming

SUBMISSION INSTRUCTIONS

Please submit your work to Week11 assignment folder

PREPARATION

Download the template project available in week11 resources folder. The solution has two projects: **DataStructures_Algorithms & Runner** projects.

LAB TASKS

1. LET'S IMPLEMENT THE 0/1 KNAPSACK PROBLEM

In this task we want to implement the 0/1 Knapsack problem using two different approaches: **Brute Force** and **Dynamic Programming**. In particular, in **Week11/BinaryKnapSackSolver.cs** you are given the **Item** class which describes items to be put in a knapsack. You are also given the **BinaryKnapSackSolver** and need to implement

a) public void BruteForce(Vector<Item> Items, int Capacity, ref Vector<Item> SelectedItems, ref int BestValue). This method receives a list of items and the capacity of the knapsack then returns a list of selected items and the best value (which is the sum of the value of selected items). Note that you will need to implement this method using brute force approach, i.e. investigate all possible cases of selected items and find the best case.

   Hint: You could first get all possible subsets of the given set of items. You are provided with the Powerset method in your assignment 1 which would help you to identify all the subsets given a set.

b) public void DynamicProgramming(Vector<Item> Items, int Capacity, ref Vector<Item> SelectedItems, ref int BestValue). This method also receives a list of items and the capacity of the knapsack then returns a list of selected items and the best value (which is the sum of the value of selected items). However, in this method, you will need to use dynamic programming. You should refer to Week 10 lecture for the implementation.

c) You are given the runner class in **Runner11_Task1.cs**. In this runner, you are provided with two test cases. You need to run both of these cases. For each test case, you need to test both the Brute Force and Dynamic Programming approach and compare their time complexity.

THANKS A LOT FOR YOUR HARDWORK THROUGHOUT THE TRIMESTER, I WISH YOU HAVE FOUND THIS UNIT AS YOU EXPECTED – CHALLENGING & USEFUL.

GOOD LUCK WITH YOUR EXAMS!