CSI - 3105 Design & Analysis of Algorithms
Fall 2019

Assignment #3

Deadline: Tuesday, November 19, 2019 before 10:00 AM (to be delivered in my hands)

In class, we studied the $[0, 1]$-knapsack problem. In this assignment you have to explain how to solve the $\{0, 1\}$-*knapsack problem* using dynamic programming.

You are given n objects which *cannot be broken into smaller pieces*. Moreover, *you have only one copy of each object*. Each object i (where $1 \leq i \leq n$) has weight $w_i > 0$ and value $v_i > 0$. You have a knapsack that can carry a total weight not exceeding W. Your goal is to fill the knapsack in a way that maximizes the total value of the included objects, while respecting the capacity constraint. For each object i (where $1 \leq i \leq n$), either you bring it or not.

1. (10 marks) Write a recursion for the optimal solution and explain why it is correct. **Make sure you define the notation you are using.**

2. (10 marks) Consider the following input and fill the table corresponding to the recursion you found in #1: $n = 6$, $w_1 = 2$, $w_2 = 3$, $w_3 = 2$, $w_4 = 9$, $w_5 = 3$, $w_6 = 2$, $v_1 = 7$, $v_2 = 1$, $v_3 = 6$, $v_4 = 18$, $v_5 = 22$, $v_6 = 28$ and $W = 11$. Moreover, give all optimal solutions.