CS 401: Part 2 of Assignment #2 (1 problem, total 50 points)

For regular students, the deadline is **April 15** in class.

For special needs students, the deadline is **April 22** in class.

No late assignments will be accepted.

Special note: Any answer that is not sufficiently clear even after a reasonably careful reading will not be considered a correct answer, and only what is written in the answer will be used to verify accuracy. No hand waiving, vague descriptions or sufficiently ambiguous statements that can be interpreted in multiple ways will be considered as a correct answer, nor will the student be allowed to add any explanations to his/her answer after it has been submitted.

Problem 1 [50 points] Consider the following **more general version** of the Knapsack problem. There are p **groups of objects** $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_p$ and a knapsack capacity W. Each object x has a *weight* w_x and a *value* v_x. Our goal is to select a subset of objects such that:

- the total weights of selected objects is at most W,
- **at most one object is selected from any group**, and
- the total value of the selected objects is *maximized*.

Suppose that n is the **total number of objects in all the groups**. Give an $O(nW)$ time algorithm for this general Knapsack problem. Explain why your algorithm is correct and analyze the running time of your algorithm.

Hint: This is very similar to the Knapsack problem seen in the class except that we have to consider groups of objects with the restriction that at most one object from any group may be selected. Let $K(w, j)$ denote the total value of all selected objects in an optimal solution when the total weight of all selected objects is at most w and only objects from groups $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_j$ are considered. Write a recurrence for $K(w, j)$ involving $K(w', j')$'s where either $w' < w$ or $j' < j$. Use the recurrence equation for the algorithm.