Week 6 HW: Knapsack Problem Implementation

Objectives:

- understand and use python unit tests
- understand and use recursion and dynamic programming in python
- experiment using LLMs to code and test
- analyze the use of LLMs to code and test

For this homework assignment you will implement the constrained knapsack problems as described below. You are allowed to use resources and LLMs but **do be sure to document their usage**.

Your comment header should contain **your name, the date, honor code statement, and typical description of usage of assistance(or state if you didn't use assistance)**. Below that you should describe your solution and problem solving steps. Write this with each section labelled (A-E). It should explain:

A. describe your solution in your own words (150-200 words)
B. describe sequential list of steps and substeps you took to solve this
C. describe what you tried, what worked, what did not work
D. describe where did you get stuck and how you tried to get unstuck
E. describe how you verified the correctness of code throughout the process

We recommend you first make sure you understand the problem and manually work through the given test cases. It's also a good idea to take notes for your description along the way as you try to solve the problem. Your final solution should be submitted to WebCAT.

Programming Assignment

Conflicting Items

Write a dynamic programming solution to this problem. As needed, search online and use LLMs to learn how to test and solve it. Submit your solution, **W6_grouped_knapsack.py** to WebCAT below after thorough testing. See sample tests below.

Problem Description:

Solve the knapsack problem when certain combinations of items conflict with each other. If you select one item from one combination, you cannot select the others. Given a set of N items, each with a cost and a value. The goal is to select a subset of these items to maximize the total value without exceeding a given a total cost C . However, the problem can have additional complexities. Write a function named **grouped_knapsack** that takes the following parameters:

- G (int): Number of combination groups.
- C (int): Capacity of the backpack in terms of cost.
- items (List[Tuple[int, int]]): A list of N items, each represented as (cost, value).
- groups (List[List[int]]): A list of G groups, each containing item indices from the `items` list.

Each item will be in one and only one combination. Combination groups can contain 1 or more items. Find out which items can be put into the knapsack so that the total cost of these items does not exceed the capacity of the knapsack and the total value is the largest. The **grouped_knapsack** function should return:

- int: The maximum total value achievable.
- List[int]: The list of selected item indices.

```
# Sample Test Case
# Description: Simple case with two groups, each with two items.
G1 = 2
C1 = 5
items1 = [ (1,2), (2,4), (2,3), (3,5) ]
groups1 = [
    [0, 1],  # Group 0: Item 0 and Item 1
    [2, 3]   # Group 1: Item 2 and Item 3
]
```

Rubric: Will be scored on WebCAT 50pts autograded and 50pts the following (labelled Design/Readability on WebCAT).

- 25pts describe your solution in your own words (150-200words)
- 5pts sequential list of steps and substeps
- 5pts what you tried, what worked, what did not work
- 5pts where did you get stuck and how did you try to get unstuck
- 10pts how did you verify correctness of code throughout the process

There will be a 5 pt deduction for excluding your name, the date, honor code statement, and typical description of usage of assistance(or state if you didn't use assistance)

AI Use Documentation Requirement

You are allowed to use resources, including Generative AI, to assist with this assignment. However, **you must explicitly document your usage**. Just below your honor statement, **you must include the following statement:**

> **During the preparation of this assignment, [name of the student] used [name of the tool] in [name the parts of the assignment where it was used] to [specify the purpose]. After using this tool, I/we reviewed and edited the content as needed to ensure its accuracy and take full responsibility for the content in relation to grading.**

Failure to include this statement may affect your grade.

Submit to WebCAT:

In vscode you can rightclick on your python file and download it to your computer. Then you can submit to WebCAT and select that python file to upload. Double check that you are uploading **W6_grouped_knapsack.py.**

There is a limit of 10 submissions. Submission energy (https://canvas.vt.edu/courses/204793/pages/web-cat-submission-energy) is also turned on. Do the best you can to test your solution locally before submitting.

Upload Your File(s)

Submission Energy

For: CS 2104 (13316 TR 8:00 - 9:15am (NIzamani)) HW6 Knapsack Grouped: HW6 Knapsack Grouped

Upload Submission (/Web-CAT/WebObjects/Web-CAT.woa/wo/2.0.0.0.29.1.9.0.1.1.1.0.0.1.9.1.1.1)

+ **Rubric**