CS 151 Lab 10: Shopping Spree

Overview

Remember that you have just tried solving a "Shopping Spree" problem:

You just won a raffle for a shopping spree at your favorite store! They hand you an empty bag that can only hold up to (and including) ***30 pounds****. You have 15 minutes to fill the bag with any items you want.*

What will you choose to maximize your bag's value?

This is actually called the "Knapsack Problem" and is typically studied in more depth in Algorithms. You probably noticed that (much like the n Queens problem), there are two subproblems that can be framed to solve the Knapsack Problem.

- Checking if a set of items fits in the bag; if so, computing its value.
- Finding the set of items that fits in the bag and maximizes (has the largest) value.

In this lab, you'll focus on the first task; in order to model the problem, you'll work with a *dictionary* to store the data required.

Learning outcomes

Practice fundamental developer skills to build a program from scratch:

- Understanding a problem and choosing how to model it.
- Creating a design for your approach.
- Implementing your design.

Practice programming skills:

- Work with *dictionaries* in Python.

Submission

1. Submit your group's **shoppingSpree.py** program on Gradescope; be sure to have all group members on the submission, but you only need one submission.
2. Using Week 10’s forum, post a reflection about your experience:
    - *How was today’s lab?*
    - *Did you find your comfort level shift for:*
        - *Building a program from scratch*
        - *Working with dictionaries*
    - *Any other observations?*

Required tools

To complete this lab, you will use:

- A web browser (e.g., Firefox, Chrome)
- VS Code

Part 1: the dictionary data structure

We have worked with lists, which hold data sequentially; to access elements, we can use their *indices*. In a way, an index is a unique identifier associated with a piece of data. For example, if we think of a list as a row of seats in a theater, then we can direct an audience member to seat 5 without them wondering where they should go. The number 5 uniquely identifies the seat.

We don't always have a natural sequential ordering associated with data. For the shopping spree, we think of our data in terms of item names (e.g., "paper ring" or "golden shoe"). Python offers a *dictionary* data structure to support data like this. You may want to think of it like a spreadsheet, where the first column holds unique *keys* and the second column holds associated *values*.

In small groups and as a class, we'll use a worksheet to practice working with dictionaries.

Part 2: shopping spree (checker)

In your groups, build a program that implements a shopping spree "checker."

Requirements

- You should use dictionaries to store the data for the store items (their prices and weights so you may want a dictionary for each mapping an item name to its price and one mapping the item name to its weight)
- You should have at least two test cases that you check:
  - one for a valid set of items (whose total weight is at or below the capacity)
  - one for a invalid set of items (whose total weight is greater than the capacity)
- Bonus: if you have time, allow the user to input the items they want to put in their bag, then print the outcome
- Bonus bonus: come up with an approach for finding the best solution for the shopping spree

  [HINT: it can be done recursively… iterate over the items and compare the bag with and without the item]

Remember the design-implementation process

- Determine the data structures and behavior for your program
- Write pseudocode that captures your design and is "close to" begin implementable in Python
- Copy your pseudocode as comments to your Python program
- Use stubs to allow you to implement one function at a time

Submission checklist

- Gradescope
  - Submit your group's **shopping_spree.py** program on Gradescope; be sure to have all group members on the submission, but you only need one submission.
- Moodle:
  - Use Week 10’s forum, post a reflection about your experience
  - *How was today’s lab?*
  - *Did you find your comfort level shift for:*
    - *Building a program from scratch*
    - *Working with dictionaries*
  - *Any other observations?*