**Problem Statement**

Introduction

For the next week, you will be learning about Data Structures and Algorithms in Python. This assignment is designed to aid your learning as you progress in these modules.

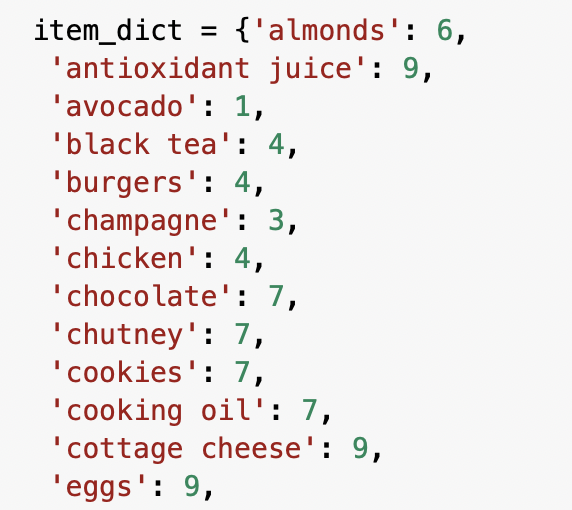
Problem Statement

You have been hired as the Tech Lead for a Software company that builds the technology behind departmental stores such as Walmart and Amazon Fresh. Your job is to use your programming knowledge to automate some of the tasks that occur in these stores. Some of these tasks include automating the billing process, building an inventory management system, and finding insights on product sales. This will help increase the efficiency at which your departmental store clients function.

Understanding the Data

You have two types of data available. The first is price data, which is available in a Python data structure called a dictionary. This dictionary has been named item\_dict, and it stores the price of each item that is available in the store.

For example, the price of almonds is $6 and the price of black tea is $4. For future reference, we will call this the Cost dictionary.



Item dictionary

The second is the Transaction data, which is available as an array of arrays. Each row corresponds to one transaction. For example, in the table below, the second transaction consists of burgers, meatballs, and eggs. For future reference, this will be called transaction data.



Transaction Data

**Data**

**Download**

Q1. Based on customer demand, items keep changing in the store. Your first task is to create a function that lets the admin add or remove items from the store directory(Cost dictionary).

* The code should help admin to accomplish the following tasks:

1. Find how many unique items are present in the data?  (Hint: It will correspond to the number of key pairs in the Cost dictionary)
2. Find the total number of transactions?
3. Add items, try adding the item “banana” in stores. The cost of this item is $5. Add the item “banana” to the Cost dictionary.
4. Use a function named *add\_item( )* to automatically add an item to the Cost dictionary given its name and price.
5. Remove an item: for example: if the customers at Walmart do not seem to be purchasing “banana,” and the store decides to discontinue this item. The function to should be able to remove the item “banana” from the Cost dictionary
   * Create a function called *remove\_item( )* to automatically remove an item from the Cost dictionary given its name.

Q2. Once a customer has completed shopping, they need to be billed for the total amount of all the items. This can be done automatically by swiping each item’s barcode on a checkout system. Your second task is to build the software underlying the checkout system, which finds the total cost of all the items in a customer’s cart, including taxes:

* Write the code, to find the price of an item from the cost dictionary
  + Find the price of salmon from the cost dictionary
* Write a code that helps the administrator find any particular transaction in the sequence
  + Try finding the ninth transaction from the data
* How many items are present in the ninth transaction?
* Find the total cost of the ninth transaction from the array of transactions. Example: Input: [‘avocado’, ‘chutney’, ‘cookies’], Output: 15
* As per government guidelines, a mandatory 5% tax is applied to the total cost of a transaction. Find the final price of the second transaction after applicable taxes have been implemented.
* Create a function named *find\_total( )*to find the final price of any transaction after applicable taxes have been implemented.

Q3. There is always a flow of items in and out of stores, which need to be tracked so that the store can place an order in case an item is running out of stock. Your third task is to build an inventory management system that will count the total number of items purchased from the transaction data. This inventory management system will go through each of the transactions one by one and increment the value for an item if it is found in a transaction.

* What is the data structure that can be used to count the frequency of items purchased (from the transaction data)?
* Create a frequency counter to count the number of items:

|  |  |
| --- | --- |
| ***Item*** | ***Frequency*** |
| Almonds | 1 |
| Antioxidant Juice | 1 |
| Avocado | 2 |



frequency\_dict = {}  # creating the empty dictionary

**for** transaction **in** transaction\_data: # looping over each transaction

**for** item **in** \_\_\_(**1**)\_\_\_\_: # looping over each item present in the transaction

**if** item **in** \_\_\_\_\_(**2**)\_\_\_\_\_:

     \_\_\_\_\_\_\_(**3**)\_\_\_\_\_\_\_\_ # increment the frequency by 1 if the item is found in the dictionary

**else**:

      \_\_\_\_\_\_(**4**)\_\_\_\_\_\_ # create a new key with the item name if the item is found for the first time

* A new transaction has just been made. This transaction includes “honey,” “pet food,” and “shrimp.” Update the frequency counter to reflect this new transaction.
* Create a function named *update\_counter( )* to automatically update the frequency counter in case any new transaction is made.

Q4. Walmart is running a discount week for Christmas! The sales manager has decided to keep a 20% discount on the top three selling items. Use the frequency counter that you created in Question 3 to solve this problem:

* What is the top-selling item?
* In real life, departmental stores have thousands of items and millions of transactions in their databases. In such situations, it becomes important to use an effective searching algorithm to navigate through the data. You need to sort the frequency counter dictionary by the value count(or the number of items). For example, if there are 3 apples, 5 bananas and 1 watermelon, the sorted list would be banana, apple, watermelon. So you need to sort the entire dictionary (with keys and values) based on the value count. Sort the frequency counter using the sorting algorithms provided in the table below. Additionally, for each of the sorting methods, leverage the time function in Python to see how many milliseconds it takes for an algorithm to sort items.

|  |  |
| --- | --- |
| ***Algorithm*** | ***Time*** |
| Merge |  |
| Insertion |  |
| Selection |  |

* Rank the sorting algorithms in order of the time they take to sort items.