CS 445 Lab 2: Bag Client

Introduction

In this lab, you will implement a brute-force algorithm for the longest common subsequence problem (LCS) using a Bag. You will be using the Bag without delving into the details of how the class is implemented; that is, you will be writing *client code*.

The primary goal of this lab is to practice solving problems using the methods provided by a data structure. A secondary goal is to familiarize yourself more deeply with the ADT Bag and its potential uses.

The ADT Bag is one of the primary collection structures defined in mathematics. Also called a multiset, it is a generalization of a set that is allowed to hold multiple copies of an item. Like a set, the items contained within the set have no particular ordering. Before completing this exercise you should review the methods available to you in the Bag ADT.

Your TA will give a lesson reviewing the ADT Bag and introducing the longest common subsequence problem.

Exercise

After the TA's lesson, complete the following steps:

1. Download the provided code, located on the course website where the following java files are provided.

- `BagInterface.java` is a Java interface representing the ADT Bag.
- `ArrayBag.java` is a dynamic capacity array-based implementation of ADT Bag. You will not need to focus on this implementation in this exercise.
- `LongestCommonSubsequence.java` provides the skeleton of a LCS solution.

Note that all of these files are in the package `cs445.lab2` . Review Lab 1 if you do not remember how to handle code in packages.

2. Review the following algorithm that finds the LCS of two input strings. This algorithm takes a brute force approach of generating all possible subsequences and checking them. Note that, while this algorithm will get the correct answer, a more efficient approach would be to use dynamic programming (a topic from CS 1501).

```
Longest Common Subsequence (first, second):
    Create an empty bag
    Put the first string into the bag
    Set the longest subsequence to the empty string
    While the bag is not empty:
        Remove a test string from the bag
        If the the longest subsequence is shorter than the test string:
            If the test string is a subsequence of the second string:
                Set the longest subsequence to the test string
            Otherwise, if test is at least 2 longer than longest subsequence:
                Generate new strings from test by removing each single charact
                Put each of the new strings in the bag
            Print the Bag of strings that need to be checked
        Print out the longest subsequence
```

Note that `LongestCommonSubsequence.java` already contains a method, `isSubsequence` , to check whether one string is a subsequence of another.

3. Once you understand the algorithm from Step 2, read through `LongestCommonSubsequence.java` , noting the 2 `TODO` comments. You will need to complete these portions of the code.
4. At the first `TODO` comment, create a new bag of strings, assign it to the variable `possibleSubsequences` , and add the string first to the bag.
5. At the second `TODO` comment, implement the algorithm from Step 2.
6. Test the program to be sure it works. Below are pairs of strings and their correct longest common subsequence.

First	Second	LCS
D	ABC	
AA	ABA	AA

First	Second	LCS
AA	AAA	AA
AABC	ABBC	ABC
ABBC	ABCC	ABC
ABCC	CABAC	ABC
ABA	AA	AA
ABC	CBACBA	AB or BC or AC
ABC	CBACBACBA	ABC
ABC	BCABCA	ABC
ABCD	DCBADCBA	AB or AC or AD or BC or BD or CD
ABFCD	ADBAFDCBA	ABFC or ABFD
ABFCD	ADBADCBA	ABC or ABD
ABCDF	ADBADCBA	ABC or ABD
ABADA	BADABAABDBA	ABADA

Conclusion

In this lab, you wrote client code using the ADT Bag to solve the LCS problem. Writing client code is a crucial skill in Data Structures. Throughout the term, you should take the time to practice writing code that uses the data structures presented. to improve your skills in solving problems using the (often limited set of) operations available to you.