CSE 104 – Data Structures & Algorithms

Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University

Assignment 1

Learning Outcomes

On successful completion of this assignment, students are expected to:

- understand and be able to apply a variety of data structures together with their internal representation and algorithms;
- be able to select, with justification, appropriate data structures to ensure efficient implementation of an algorithm.

To Do

Using the techniques learnt from our Data Structures & Algorithms module, develop in Java a **subclass of List** with **type parameter** that includes the methods described below. Specifically, develop an algorithm for each method. Finally, write in Java a program that implements the following methods and a **main** method that in order to test these methods.

Tasks

1. Remove all target instances from a list.

Design a method which takes **ONE** parameter: ***target***, remove all instances of the target and **return** the new length. The order of elements can be changed. It doesn't matter what you leave beyond the new length.

Example 1: You have a list A= [6, 2, 1, 3, 2], input a target 2. You have to remove all the 2 from A, and return the new length that matters, in this case is 3.

Example 2: You have a list B= ["accordion", "clarinet", "banjo", "clarinet", "clarinet", "cLarinet"], input a target "clarinet". You have to remove all "clarinet" from B, and return the new length that matters, in this case is 3.

Note, "Clarinet" or "cLArinet" are not equal to "clarinet".

2. Remove items from a list that appear three times.

Design a method which takes **NO** parameter, remove the items that appear three times, and print out the new list and **return** the length of the new list.

Example 1: You have a list A = [1, 1, 1, 2, 2, 3]. Your function should remove all the items that appear three times in the list, and print out the new list. In this case the print out is: [2, 2, 3]. Also, you function should return 3 which is the length of the new list.

Example 2: You have a list B = ["accordion", "banjo", "clarinet", "banjo", "clarinet", "clarinet", "Banjo"], You function should remove all the items that appear three times in the list, and print out the new list. In this case the print out is: ["accordion", "banjo", "banjo", "Banjo"]. Also, your function should return 4 which is the length of the new list.

Note, "Banjo" or "BanJO" are not equal to "banjo".

3. Find all sets of strings that are anagrams.

An anagram is a word or phrase formed by **rearranging** the letters of a different word or phrase, typically using all the original letters **exactly once**.

Design a method which takes **NO** parameters and **return** all the sets of anagrams in this list.

For example: You have a list of strings A = [“tea”, “eat”, “acde”, “ate”, “adec”, “cdf”, “cdf”, “tea”, “adecc”], your method should find all the sets of anagrams and return them. In this case, your method should return all the sets: both [“tea”,”eat”,”ate”] and [“acde”,”adec”].

Note, duplicates are not considered as anagrams, “cdf” and “cdf” are duplicates, and they are not considered as anagrams. These strings will only contain letters, you don’t have to worry about other characters like space or ‘,’ or any others. You don’t have to worry about cases in this method. Consider all your strings are lower cases.

Due date

2019-04-19 12:00pm (Submit on ICE)

What to Submit

- One or two .java file.
- A simple report which contains:
    - A XJTLU front page with module title, student name/number, declaration of non-plagiarism etc.
    - One to two page algorithms pseudocode.
    - One to two page snapshot or your testing and/or debugging.

What to do during the assessment hour during the due date?

a) Attend on-spot test.

b) The test question will be a very small modification of one of the questions above. **Modify** your own code to achieve the updated question. Question will be released on spot.

c) Submit your updated code to ICE on spot.

d) Bring your hardcopy (Stapled) and place them in your desk, a TA will come to collect them and ask your to sign the attendance sheet.

Marking

Criteria of marking: correctness, conciseness, quality and efficiency of the data structures/algorithms used/developed.

NOTE:

- Being absent of on-spot test will cause direct failure in this assignment.
- Submission of .java file(s) with compiling error will cause marks deduction. Please makes sure you test your methods before submission.
- Comments, code readability and etc. will be considered in “code quality” check.
- Plagiarism will cause direct failure in this course.
- This assignment: 10% of the overall marks for CSE104.