

Programming Assignment 4

Secret Messages

Due: 11/13/2022 at 11:59pm

Objective: Students will apply concepts of dynamic programming and the LCS problem.

Assignment Description: Your friend has intercepted a message that is riddled with random characters. Your friend tries really hard to decipher the message (“qwadhyuittrejghfhgkllqwrto”), but couldn’t. A couple of minutes later your friend receives another message (“mpoihselyzmxcvldfiuoydmnbv”) riddled with random characters. Something interesting your friend noticed is that it seems to have the same number of characters. Your friend tells you about this and you are curious too. You have just listened to a lecture by Dr. Steinberg about finding longest common sequences and immediately practice on this long message to study for giggles. After completing by hand the steps of finding the longest common sequence, you get an actual word (“hello”) as the longest common sequence and realize the secret of this message is through the longest common subsequence. The next day you tell your friend what you discovered. Coincidentally your friend receives another two message with the same length of characters. However, this one is a long sequence and can be time consuming to do by hand. In this assignment you are going to reveal the secret message. In this assignment, you have intercepted the document and are going to decipher the message. In order to do this, you will need to apply the Longest Common Subsequence Algorithm discussed in class. You must apply the dynamic programming solution. Using another variation will result in losing points.

For this assignment, you must follow these **requirements**.

1. Create a class called `LCS`. In this class, you will implement the dynamic programming algorithm.
2. The class constructor for `LCS` takes two string objects.
3. In the runner file, you will notice the method “`lcsDynamicSol`” is invoked. This is the method that will invoke the dynamic programming solution. Make sure the method name matches as the runner file in your implementation! Changing names will cause the runner file not to run or even compile which will result in a low score on the assignment!
4. In the runner file, you will notice the method “`getLCS`” is invoked. This is the method that will access the subsequence computed by the dynamic programming solution. This is the Print-LCS algorithm we discussed in class. Make sure the method name matches as the runner file in your implementation! Changing names will cause the runner file not to run or even compile which will result in a low score on the assignment!
5. Make all methods public and class attribute private. It’s good practice!
6. You may create additional helper methods and attributes if needed as long as they are implemented and called in your solution file and NOT called from the runner file. Adding extra methods to call in the runner file will not match to what the graders will use to evaluate your code. This will result in a low score with no change to be applied!

7. Your code must run within 2 seconds on Eustis. If it runs longer than 2 seconds, an automatic score of 0 will be given for the assignment overall.

A runner file (`LCSRunner.java`) has been provided for you to show you how the methods are called along with 4 test cases. The text file is also provided for you that you will need to decipher using the LCS algorithm. The text file itself must be in the same directory as the runner file. The number of characters to be read from the file on each line will have no more than 1000 characters.

What to submit: Submit a file called `LCS.java` to webcourses. You are not required to submit the runner file as that will be provided for the graders to test your code. Please make sure the runner file provided works for your code. Any name changes may cause your program not to work when graded, which will result in a lower score on the assignment and would not be changed.

Important Note for running Eustis: Many of you are probably using IDEs like Netbeans and Eclipse to build your Java Solutions. Please note that some of these IDEs will automatically place your Java file in some sort of package. Please make sure your Java file is not defined in some package as this can result package private errors. Any such error that occurs during the grading will not be fixed and points will be deducted as such in accordance with the respective categories in the rubric. Also, **DO NOT** create a main method in your solution file!! This will result in your code not running properly with the runner file which will result in points being deducted from the respective categories.