EE 360C — Algorithms — Summer 2013
Programming Assignment #4

Due: Aug 14, 2013 11:59pm (via Blackboard)

Programming assignments are to be done individually. You may discuss the problem and general concepts with other students, but there should be no sharing of code. You may not submit code other than that which you write yourself or is provided with the assignment. This restriction specifically prohibits downloading code from the Internet.

Problem Description

A lot of data occurs in the form of sequences. A *sequence* is defined as an ordered list of items where an item can be repeated multiple times in the list. For example, a strand of DNA consists of four different bases: Adenine, Cytosine, Guanine, and Thymine. These bases are usually represented by their first characters, and thus a strand of DNA can be expressed as a string consisting of the following set of characters: {`a`, `c`, `g`, `t`}. In computational biology, it is useful to compare DNA strings for similarity. When comparing two DNA strings, exact matching is not always important. An exact matching algorithm can only tell you if two DNA strings are equal or not. Very often, it is useful to have a measure of similarity that is not binary.

In this programming assignment, we will study a dominant measurement of similarity between sequences: *longest common subsequence* (LCS). Note that the items in a sequence can be any abstract objects but in this programming assignment we will assume that the sequences are strings and the items are thus characters.

Longest Common Subsequence

A *subsequence* of a given string is defined as that given string with zero or more elements deleted and the LCS of two strings S_1 and S_2 is defined as the longest subsequence that is a subsequence of S_1 as well as a subsequence of S_2.

In simple terms, the LCS is the string that is left over after you have applied the minimum number of deletions to transform the two strings into a common subsequence. Note that a common subsequence can skip some characters as long as the relative ordering of the characters is always preserved. For example, let S_1 = `agttgtagct` and S_2 = `agtgctact`. The LCS of S_1 and S_2 will be `agtgtact` and note that it appears in both S_1 and S_2 in order: S_1 = **agt t gta g ct** and S_2 = **agtg c tact**.

Grading

Grading for this assignment is based on accuracy of the data output (longest common sequence string), running time (if you implement brute force, you will get zero marks as the running time will be exponential and the grader will timeout for those tests with larger test inputs) and compliance to the instruction (25% deduction for non compliance and no score for compilation error as we simply do NOT have time for this assignment to manually change your program to make it work).

Input and output specification

Input will be two given on standard input as two strings on separate lines. If you still have problems reading standard input as it turned out to be the case in lab1 and lab2, please check the clarification email sent by the instructor. For example, given the following input:

```
agttgtagct
agtgctact
```

the output should be the longest common sequence i.e. `agtgtact`. There is no space among characters and no beginning/ending space.

Submission Instructions

- Make sure your program compiles on LRC machines before you submit it. It should compile using the standard commands `javac *.java`, `gcc *.c`, or `g++ *.cc` without any extra switches or any additional libraries. If you use windows especially Microsoft Visual Studio, make sure your header files are generic and not Microsoft specific. You will get zero marks if you use non-generic header files.
- You should submit a single zip file titled `eid_lastname_firstname.zip` that contains all your program files and optionally a readme file. Do not put these files in a folder before you zip them (i.e. the files should be in the root of the ZIP archive) and do not include binaries.
- Please name your main program file as Lab4 (in Java Lab4.java, in c++ Lab4.cpp), failure to do so is treated as compilation error.
- Your solution must be submitted via Blackboard *before* the deadline. No late submissions will be accepted.