CS 310 — Algorithms — Fall 2020
Programming Assignment #4

Due by 11pm on 6 November 2020

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, you will implement 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**.

You must implement this algorithm in polynomial running time.

Input and output specification

Input will be given on standard input as two strings on separate lines. For example, given the following input:

```
agttgtagct
agtgctact
```

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