

CS1550 Algorithmic design

Tom Bitterman, David Mestel

Assignment 2. Deadline: 23:59, Wednesday 24 April

1 Introduction

You need to submit your solutions via Codegrade, which can be reached from the assignment page on Canvas. Your program needs to read a filename from the terminal, read the input from the named file, and print the output (and nothing else) to the terminal. The Canvas page for each task contains an example `.java` file demonstrating the I/O. Note that **the file you upload must have the same name and main class as the example, otherwise the grading will fail**. An example input file is also included, and its expected output. You can submit as many times as you want, subject to a limit of one submission every 5 minutes.

The deadline is 23:59 on Wednesday 24 April, and this deadline is strict. **You are therefore strongly encouraged to submit solutions well before the deadline, in case of technical problems. Technical problems with submitting to Codegrade will not be considered a reason to extend the deadline.** The runtime time limit for each test case is 20 seconds. You must not try to exploit the automatic grader in any way, or attempt to exfiltrate the test input data—doing so will be considered cheating.

2 ‘Anchored’ common subsequences

Remember the definition of longest common subsequence from the lectures: the longest common subsequence of strings s_1, s_2 is the longest string s such that s is a subsequence of both s_1 and s_2 , i.e. we can get s by starting at some point in s_i , reading through, perhaps skipping some letters, and then stopping at some point.

Suppose now our alphabet contains a special ‘anchor’ character `*` which binds ‘more strongly’ than ordinary letters: concretely, when reading through each s_i to find s we are not allowed to skip any `*`s (although there may be `*`s outside the section of s_i we read).

So for example, suppose that $s_1 = A * BCD*$ and $s_2 = B * CAD$. We would *not* be allowed to take the subsequence `BC` (because this requires skipping a `*` in s_2), but we *would* be allowed to take the subsequence `*CD`, which is the longest anchored common subsequence and has length 3.

The input file will have two lines, each consisting of a string over alphabet $A, B, \dots, Z, *$ of length at most 2,000 characters. Your program should output a single integer n , the length of the longest anchored common subsequence.

Name your program `Anchored.java` for submission to Codegrade.

3 Examples

$s_1 = A * BF * CB$
 $s_2 = F * CAB * D$
 LACS = $F * CB$

$s_1 = A * B * CD*$
 $s_2 = B * CAD$
 LACS = $B * CD$

$s_1 = BFA * B$
 $s_2 = CB * A * BB$
 LACS = $A * B$

$s_1 = *AC*$
 $s_2 = *BD*$
 LACS = $**$

$s_1 = ACCBDFBA * BA$
 $s_2 = CBAAF * BAA$
 LACS = $CBF * BA$ or $CBA * BA$

4 Hints

Try to understand how to solve the problem by hand before you start coding. Coding up the new algorithm is not the hard part. Figuring out what the algorithm is, is.

To do this, start with the array algorithm for Longest Common Subsequence (it was presented in class). Be able to do this by hand for strings that do not contain $*$.

Then, try the algorithm out on strings with $*$.

It will not work. Try to figure out how to modify the algorithm so it will work. What is the complexity of your algorithm? Given that the input can have size up to 2,000, you are looking for an $O(n^2)$ algorithm.

Test out your new algorithm on a few more cases you make up. Start with small cases where you can see the answer by inspection (just looking at the problem) and which are easy to work out by hand. Continue trying different

cases until you cannot think of a case where your algorithm gives the wrong answer.