

Algorithmics - Tutorial Sheet 3

Strings and text algorithms

1. **[Work in pairs]** Find the distance between the strings s and t shown below by drawing the complete table filled in based on the recurrence relation defined in Lecture 3, and then derive an optimum alignment.

$s = \text{agcgatc}$ and $t = \text{ctacgaccg}$

2. A string u is a subsequence of a string s if u can be obtained from s by deleting zero or more characters. A string u is a common subsequence of s and t if it is a subsequence of both s and t . (The length of an LCS is often used as a measure of similarity of two strings.)

Design a dynamic programming algorithm to determine the length of the longest common subsequence (LCS) of two strings s and t .

Hint: base your algorithm on evaluating $l(i, j)$, the length of the LCS of the i th prefix of s and the j th prefix of t .

3. **[Work in pairs]** Find a longest common subsequence of the strings s and t in Question 2 above.