PSO 2 — Week 5: Problems

CS381, Fall 2026

Problem 1. Common subsequences

Let $A[1..n]$ and $B[1..m]$ be strings, and write $L(A, B)$ for the length of a longest common subsequence of A and B.

(a) Let $L[i][j]$ be the length of a longest common subsequence of $A[1..i]$ and $B[1..j]$. State a recurrence for $L[i][j]$ and prove it. In the case $A[i] = B[j]$, checking that matching $A[i]$ with $B[j]$ is one available option only proves $L[i][j] \geq L[i-1][j-1] + 1$; ruling out anything longer takes an exchange argument. Then show how to read an actual longest common subsequence off the finished table in $O(n + m)$ time, storing nothing beyond the table itself. Finally, a classmate computes the longest common subsequence of three strings as $\mathrm{LCS}(\mathrm{LCS}(A, B), C)$: give a triple on which this is wrong, and say what information the composition throws away.

(b) Let $d_{\mathrm{id}}(A, B)$ be the minimum number of single-character insertions and deletions turning A into B, with no substitutions allowed. Prove that

$$d_{\mathrm{id}}(A, B) \;=\; n + m - 2\,L(A, B).$$

Then, writing $d(A, B)$ for the edit distance from lecture, show $d \leq d_{\mathrm{id}} \leq 2d$ and give a family of pairs on which the second inequality is an equality, so that $n + m - 2L$ is exactly twice the edit distance. Finally, write the recurrences for d_{id} and d side by side and identify the one place where they differ.

(c) *(harder)* Suppose the characters of A are pairwise distinct; B is arbitrary. Show that $L(A, B)$ can be computed in $O(n + m \log m)$ time, beating the table, using the patience-sorting routine from Week 4, Problem 1(c) as a black box. Prove the correspondence your reduction relies on in both directions, and point out the one step that needs more than the ability to compare two characters. Then say what breaks when A has repeated characters, and why the natural repair (replace each character of B by the list of its positions in A) is not subquadratic in general.