

Quiz 8 S22 SOLUTION

Due Date Thursday Nov 10, 2022 8pm MT
Name **Your Name**
Student ID **Your Student ID**
Quiz Code (enter in Canvas to get access to the LaTeX template) **IKJHG**

Contents

Instructions	1
Honor Code (Make Sure to Virtually Sign)	2
22 Standard 22: Dynamic Programming: Write down recurrences	3

Instructions

- You may either type your work using this template, or you may handwrite your work and embed it as an image in this template. **If you choose to handwrite your work, the image must be legible, and oriented so that we do not have to rotate our screens to grade your work.** We have included some helpful LaTeX commands for including and rotating images commented out near the end of the LaTeX template.
- You should submit your work through the **class Gradescope page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign)

Problem HC. • My submission is in my own words and reflects my understanding of the material.

- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (signature here).



22 Standard 22: Dynamic Programming: Write down recurrences

Problem 22. Consider a modified version of the string alignment (aka edit distance) problem that we saw in class. The allowed operations are: insert, delete, substitute, and *swap*. **Note that the swap operation is new compared to what we covered in class.** (Also, a no-op, which is free, as in class.)

The **swap operation** replaces any two adjacent letters by swapping them. For example, performing a swap on *there* could result in *htere* (by swapping the first two letters), *tehre* (by swapping the second two letters), *three* (by swapping the *er* in the middle), or *theer* (by swapping the last two letters).

Given two strings x and y , let

$T[i, j]$ = minimum number of operations to transform the prefix $x[1..i]$ into the prefix $y[1..j]$.

Your job is to write down a recurrence for this dynamic programming table. Make sure to include your base cases and justify your recurrence.

As a starting point, we have included the recurrence from class for the simpler version of string alignment we did in class; if you choose to start from this recurrence, you will have to edit it appropriately.

Answer. The recurrence is:

$$T[i, j] = \begin{cases} i & j = 0 \\ j & i = 0 \\ T[i-1, j-1] & x_i = y_j \\ 1 + T[i-2, j-2] & x_i = y_{j-1} \text{ and } y_j = x_{i-1} \\ 1 + \min\{T[i, j-1], T[j-1, i], T[i-1, j-1]\} & x_i \neq y_j \end{cases}$$

The only difference from the recurrence in class is the inclusion of the fourth possibility, which corresponds to the use of the swap operation. If the last two characters of $x[1..i]$ are the same as those of $y[1..j]$ but in the reverse order, then the last operation can be a swap. After that swap, we've turned the last two characters of $x[1..i]$ into those of $y[1..j]$, so we need only recurse on the $i-2$ and $j-2$ prefixes. The rest of the reasoning is the same as in class.

To see that this recurrence is valid, we need to know that, in the above case, the swap is in fact the best option. If $x_i = x_{i-1} = y_j = y_{j-1}$, then we're better off with a no-op, which is handled in the third case. Otherwise, if a swap is allowable, but we chose not to perform the swap, we'd instead have to use at least two operations to handle those last two characters, which would cost 2 instead of 1, so the swap would be better.

THE ARE ALSO OTHER WAYS TO WRITE AN EQUIVALENT RECURRENCE, E.G. ROLLING THE SWAP INTO THE min OPERATION WHEN IT'S ALLOWED. \square