CSCI 3104 Spring 2023
Instructors: Chandra Kanth Nagesh and Prof. Ryan Layer

Quiz 22 Standard 22 – DP – Write down recurrences

Due Date ..... TODO
Name ..... **Blake Raphael**
Student ID **109752312**
Quiz Code (enter in Canvas to get access to the LaTeX template) ..... **IKJHG**

Contents

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 (I agree to the above, Blake Raphael). □

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

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

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. **A version of recurrence from class. This is not a correct answer to this quiz question, we are just providing it as a starting point.**

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

Base Case: If either string is empty, then the minimum number of required edits will be the length of the non-empty string.

Case 1: If $x_i = y_j$ and the characters are not adjacent, then you simply recurse and do not need to make any edits.

Case 2: If $x_i = y_j$ and the characters are adjacent i.e. $i = j+1$ or $j = i+1$, then you need to recurse and subtract either i or j by one more since the addition of the swap edit has already been made. If we do not subtract by more, we count our swap as indel of cost 2 rather than swap at cost 1.

Case 3: If $x_i \neq y_j$, we know we are going to have an edit operation, so we add 1 and then recurse on the minimum cost of either subtracting j by 1, i by 1, or both by 1.

□