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

Final Standard 21 - Dynamic Programming: Identify the precise subproblems

Due Date ..... May 9th
Name ..... **Blake Raphael**
Student ID **109752312**
Quiz Code (enter in Canvas to get access to the LaTeX template) ..... **LDOYx**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas 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.

2 Standard 21 - Dynamic Programming: Identify the precise subproblems

2.1 Problem 1

Problem 1. Considering the sequence alignment problme, what precise sub-problems do we need to consider? How do we combine them?

Answer. Let $T(x, y)$ be the table that denotes the number of changes we need to make to get the string N_x to string M_y. That is we need to consider the number of ways to get from the current index i of N to the current index j of M for each $i \in \{1, 2, ..., x\}$ and $j \in \{1, 2, ..., y\}$. That is we consider $T(x-1, y-1), T(x-1, y-2), T(x-2, y-1), ..., T(x-i, y-j)$.
We combine the sub-problems through a recursive structure where we work backwards through the sequence in order to either find a shortest path or longest path or whatever we are searching for to align sequence N_x to M_y. □