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

Final Standard 25 - Dynamic Programming: Design a Dynamic Programming Algorithm

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

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 25 - Dynamic Programming: Design a Dynamic Programming Algorithm

2.1 Problem 1

Problem 1. Use dynamic programming to solve the the Fibonacci numbers where $F(0) = 0$, $F(1) = 1$, and $F(n) = F(n-1) + F(n-2)$ for $n > 1$. Write down the mathematical recurrence, descibe a bottom-up algorithm to solve the problem, then work throuh and example for the the following $n = 10$.

Answer. Here is the recurrence relation for the Fibonacci sequence:

$$F(n) = \begin{cases} 0 & : n = 0, \\ 1 & : n = 1, \\ F(n-1) + F(n-2) & : n > 1. \end{cases}$$

A bottom up algorithm we can use starts with $F(n)$. We can describe $F(n) = F(n-1) + F(n-2) = F((n-1)-1) + F((n-2)-2) = ...$ all the way until we reach a base case of $F(n-1) = 1$ and $F(n-2) = 0$. This is a basic recursive algorithm that will continue until we hit our base cases.

Here is working through an example for $n = 10$:

$$\begin{aligned} F(10) &= F(9) + F(8) \\ &= F(8) + F(7) + F(7) + F(6) \\ &= F(7) + F(6) + F(6) + F(5) + F(6) + F(5) + F(5) + F(4) \\ &= F(6) + F(5) + F(5) + F(4) + F(5) + F(4) + F(4) + F(3) + F(5) + F(4) + F(4) + F(3) + F(4) + F(3) + F(3) + \\ &= F(5) + F(4) + ... + F(1) + F(0) \\ &= F(5) + F(4) + ... + 1 + 0 \\ &= 34 \end{aligned}$$

The recursive structure will provide us with 34 1s that adds all up to 34 or the 10th Fibonacci number. The recurrence will keep going until all the $F(n)$ are either 1 or 0. When they hit that base case, they can add all the 1s and 0s together to get the Fibonacci number.

□