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

Homework 22

Due Date .... Apring 6, 2023
Name .... **Blake Raphael**
Student ID **109752312**
Collaborators .... **Alex Barry, Brody Cyphers, and Ben Kohav**

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 Gradescope page** only (linked from Canvas). 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 are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. 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, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

- 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). $\square$

3 Standard 22 – Dynamic Programming: Write Down Recurrences

3.1 Problem 1 (2 Points)

Problem 1. Suppose we have an m-letter alphabet $\Sigma = \{0, 1, \ldots, m-1\}$. Let W_n be the set of strings $\omega \in \Sigma^n$ such that ω does not have 00 as a substring. Let $f_n := |W_n|$. Write down an **explicit recurrence for f_n, including the base cases.** Clearly justify each recursive term.

Answer.

$$f_n = \begin{cases} 1 & : n = 0, \\ m & : n = 1, \\ (m-1)f_{n-1} + (m-1)f_{n-2} & : \text{else.} \end{cases}$$

For base case of 1, when we have $n = 0$, there is only one string to consider, and it is not 00, so we have size 1. When $n = 1$, we only have one instance of $\Sigma = \{0, 1, \ldots, m-1\}$ to consider, which does not have 00, so we have m. Finally, when we have $n > 1$, we have n of $\Sigma = \{0, 1, \ldots, m-1\}$ to consider, so we subtract the 00 out of however many strings we are considering, so we get $(m-1)f_{n-1} + (m-1)f_{n-2}$.

□

3.2 Problem 2 (2 Points)

Problem 2. Suppose we have the alphabet $\Sigma = \{x, y\}$. For $n \geq 0$, let W_n be the set of strings $\omega \in \{x, y\}^n$ where ω contains yyy as a substring. Let $f_n := |W_n|$. Write down an explicit recurrence for f_n, including the base cases. Clearly justify each recursive term.

Answer. When considering the alphabet $\Sigma = \{x, y\}$ and how many strings contain the subset yyy, we have one base case to account for and that is when $n \leq 2$. This is because it is impossible to have the string yyy with a length of 2.

Now we consider our recursive cases:
Case 1: Where the nth character of ω_n is x.

We have a string ω which is $(\omega_1), (\omega_2), ..., (x)$ or ω_n.
If x is the last letter of our string, then we only have to consider sub-strings to the left of our final letter, or f_{n-1}.

Case 2: Where the nth and $n-1$th character of ω_n are xy.

We have a string ω which is $(\omega_1), (\omega_2), ...(x)$ or $\omega_{n-1}, (y)$ or ω_n.
If xy are the last letters of our string, then we only have to consider sub-strings to the left of our final letters, or f_{n-2}.

Case 3: Where the nth, $n-1$th and $n-2$th characters of ω_n are xyy.

We have a string ω which is $(\omega_1), (\omega_2), ...(x)$ or $\omega_{n-2}, (y)$ or $\omega_{n-1}, (y)$ or ω_n.
If xyy are the last letters of our string, then we only have to consider sub-strings to the left of our final letters, or f_{n-3}.

Case 4: Where the nth, $n-1$th and $n-2$th characters of ω_n are yyy.

We have a string ω which is $(\omega_1), (\omega_2), ...(y)$ or $\omega_{n-2}, (y)$ or $\omega_{n-1}, (y)$ or ω_n.
If yyy are the last letters of our string, then we have found yyy. Next, we have to consider sub-strings to the left of our final letter to see if there are any combinations that creat yyy. This is equal to 2^{n-3} because we are considering all combinations of ω from $n-3$ and only have 2 letters to choose from.

Altogether, we have the following:

$$f_n = \begin{cases} 0 & : n \leq 2, \\ f_{n-1} + f_{n-2} + f_{n-3} + 2^{n-3} & : \text{else.} \end{cases}$$

□