CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Quiz 14 - Write Recurrence

Due Date .... March 11
Name .... **Your Name**
Student ID .... **Your Student ID**

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 14 - Writing Recurrences

Problem 1. Write down a recurrence for the runtime complexity of this algorithm. Clearly justify your answer. You are **not** being asked to solve the recurrence.

Algorithm 1 Recurrences

```
procedure Foo1(Integer $n$)
    if $n < 4$ then return 0
    Foo1($n/4$)
    Foo1($n/4$)

    for $i \leftarrow 1; i \leq 3 * n; i \leftarrow i * 2$ do
        print $i$
```

Answer.

$$T(n) = \begin{cases} O(1) & n < 4 \\ 2T(n/4) + \Theta(\log_2(3n)) & n \geq 4 \end{cases}$$

When $n < 4$, the first `if` branch is taken, so the algorithm immediately returns, using only a constant number of atomic operations. When $n \geq 4$, the algorithm makes 2 recursive calls to itself with input $n/4$, resulting in the $2T(n/4)$ term. The non-recursive work is all in the loop at the end. It has an upper bound of $3n$, and it multiplies the loop index i by 2 each time, so the loop runs $\log_2(3n)$ times. Each iteration of the loop executes a constant number of atomic operations, so the total non-recursive work is $\Theta(\log_2(3n)) = \Theta(\log n)$.

□