

## Quiz- Standard 16

---

Due Date ..... TODO  
Name ..... **John Blackburn**  
Student ID ..... **Jobl2177**

### Contents

<b>1 Instructions</b>	<b>1</b>
<b>2 Honor Code (Make Sure to Virtually Sign)</b>	<b>2</b>
<b>3 Standard 16- Analyzing Code III: Writing Recurrences</b>	<b>3</b>

### 1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to L<sup>A</sup>T<sub>E</sub>X.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this L<sup>A</sup>T<sub>E</sub>X 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 2). Failure to do so will result in your assignment not being graded.

## 2 Honor Code (Make Sure to Virtually Sign)

### Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

*Agreed John Blackburn.*

□

### 3 Standard 16- Analyzing Code III: Writing Recurrences

**Problem 2.** 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

---

```
1: procedure Foo1(Integer  $n$ )
2:   if  $n < 5$  then return 0
3:   Foo1( $n/5$ )
4:   Foo1( $n/5$ )
5:   Foo1( $n/5$ )
6:
7:   for  $i \leftarrow 1; i \leq 2 * n; i \leftarrow i * 3$  do
8:     print ( $i + j$ )
```

---

*Answer.* The base case is of constant time and is invoked everytime  $n < 5$ . This follows from the first if statement. For the non base cases IE when  $n \geq 5$ , the recursive work done by the function is 3 calls to Foo1( $n/5$ ) and the non recursive work done is the loop. The work done by the loop runs at  $\Theta(\log(n))$ . So the recurrence is:

$$T(n) = \begin{cases} O(1) & : n < 5, \\ 3T(n/5) + \Theta(\log(n)) & : n \geq 5. \end{cases}$$

□