

## Problem Set 5

---

Due Date ..... March 1  
Name ..... Alex Ojemann  
Student ID ..... 109722375  
Collaborators .....

### Contents

Instructions	1
1 Standard 14 - Analyzing Code III: (Writing down recurrences)	2
1.1 Problem 1(a)	2
1.2 Problem 1(b)	3
2 Standard 15 - Unrolling	4
3 Standard 16 - Tree Method.	6

### Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Useful links and references on  $\text{\LaTeX}$  can be found here on Canvas.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this  $\text{\LaTeX}$  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.

# 1 Standard 14 - Analyzing Code III: (Writing down recurrences)

**Problem 1.** Write down the recurrence relation for the runtime complexity of these algorithms. **Don't forget to include the base cases.**

## 1.1 Problem 1(a)

---

**Algorithm 1** Writing Recurrences 1

---

```
1: procedure Foo(Integer  $n$ )
2:   if  $n \leq 3$  then return
3:
4:   Foo( $n/3$ )
5:   Foo( $n/3$ )
6:   Foo( $n/3$ )
7:
8:   for  $i \leftarrow 1; i \leq 2 * n; i \leftarrow i + 1$  do
9:     print "Hi, Hi"
```

---

*Answer.* The comparison statement on line 2 takes one unit of time. The return statement on line 2 takes one unit of time if the comparison statement is true.

The recursive calls on lines 4, 5, and 6 each take  $T(n/3)$  units of time.

Initializing the loop on line 8 takes one unit of time.

The loop on line 8, which runs  $2n$  times, takes one unit of time to compute  $2*n$ , one to compare that to  $i$ , one to compute  $i+1$ , one to assign that to  $i$ , and one for the print statement on line 9 for a total of  $5*2n = 10n$  units of time.

We have that  $T(n) = 2 = \Theta(1)$  if  $n \leq 3$  and  $T(n) = 3T(n/3) + 10n + 1 = 3T(n/3) + \Theta(n)$  if  $n > 3$ .  $\square$

## 1.2 Problem 1(b)

---

**Algorithm 2** Writing Recurrences 2

---

```
1: procedure Foo(Integer  $n$ )
2:   if  $n \leq 4$  then return
3:
4:   Foo( $n/4$ )
5:   Foo( $n/4$ )
6:   Foo( $n/3$ )
7:
8:   for  $i \leftarrow 1; i \leq n; i \leftarrow i * 2$  do
9:     print "Hi, Hi"
```

---

*Answer.* The comparison statement on line 2 takes one unit of time. The return statement on line 2 takes one unit of time if the comparison statement is true.

The recursive calls on lines 4, 5, and 6 each take  $T(n/4)$  units of time.

Initializing the loop on line 8 takes one unit of time.

The loop on line 8, which runs  $\lfloor n^{1/2} + 1 \rfloor$  times, takes one unit of time to compare  $i$  to  $n$ , one to compute  $i*2$ , one to assign that to  $i$ , and one for the print statement on line 9 for a total of  $4*\lfloor n^{1/2} + 1 \rfloor$  units of time.

We have that  $T(n) = 2 = \Theta(1)$  if  $n \leq 4$  and  $T(n) = 3T(n/4) + 4*\lfloor n^{1/2} + 1 \rfloor + 1 = 3T(n/4) + \Theta(n^{1/2})$  if  $n > 4$ .  $\square$

## 2 Standard 15 - Unrolling

**Problem 2.** Consider the following recurrences and solve them using the unrolling method (i.e. find a suitable function  $f(n)$  such that  $T(n) \in \Theta(f(n))$ ).

(a)

$$T(n) = \begin{cases} 2 & : n < 3, \\ 2T(n-3) + 2 & : n \geq 3. \end{cases}$$

*Answer.* We unroll until  $k = \lceil (n-3)/3 \rceil$ .

$$\begin{aligned} T(n) &= 2T(n-3) + 2 \\ &= 2[2T(n-6) + 2] + 2 \\ &= 4T(n-6) + 2 * 2 + 2. \end{aligned}$$

The non recursive work each time we unroll is  $2^{i+1}$

$$\begin{aligned} T(n) &= (\text{cost of base case})(\text{number of times the base case is reached}) + \sum_{n=1}^{\lceil (n-3)/3 \rceil} 2^{i+1} \\ &= 2 * 2^{\lceil (n-3)/3 \rceil} * \sum_{n=1}^{\lceil (n-3)/3 \rceil} 2^{i+1} \\ &= 2^{\lceil (n-3)/3 \rceil + 1} + 2^{\lceil (n-3)/3 \rceil + 2} - 1 \\ &\in \Theta(2^{n/3}) \end{aligned}$$

□

(b)

$$T(n) = \begin{cases} 3 & : n < 2, \\ T(n-2) + 4n & : n \geq 2. \end{cases}$$

*Answer.* We unroll until  $k = \lceil (n-2)/2 \rceil$ .

$$\begin{aligned} T(n3i) &= T(n2(i+1)) + 4n. \\ &= T(n2(i+2)) + 8n \end{aligned}$$

The non recursive work each time we unroll is  $4n$

$$\begin{aligned} T(n) &= (\text{cost of base case})(\text{number of times the base case is reached}) + \sum_{n=1}^{\lceil (n-2)/2 \rceil} 4n \\ &= 3 * 2^{\lceil (n-2)/2 \rceil} + \sum_{n=1}^{\lceil (n-2)/2 \rceil} 4n \\ &= 3 * 2^{\lceil (n-2)/2 \rceil} + 4n * \lceil (n-2)/2 \rceil \\ &\in \Theta(2^{n/2}) \end{aligned}$$

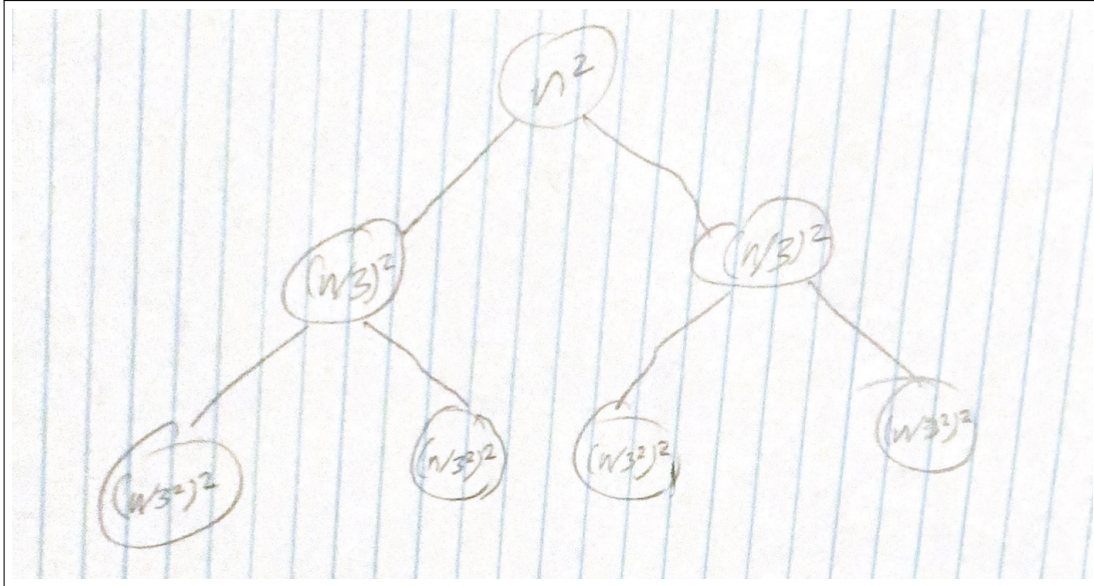
□

### 3 Standard 16 - Tree Method.

**Problem 3.** Consider the recurrence  $T(n)$  below. Using the **tree method**, determine a suitable function  $f(n)$  such that  $T(n) \in \Theta(f(n))$ . Clearly show all steps. Note the following:

- You may assume, without loss of generality, that  $n$  is a power of 3 (i.e.,  $n = 3^k$  for some integer  $k \geq 0$ ).
- You may hand-draw your tree and embed it, provided it is legible and we do not have to rotate our screens to read it. However, **all your calculations must be typed**.

$$T(n) = \begin{cases} 2, & n < 3 \\ 2T(n/3) + n^2, & \text{otherwise.} \end{cases}$$



Answer.

$$\begin{aligned} n/3^k &= 3 \\ n &= 3^{k+1} \\ k &= \log_3 n - 1 \end{aligned}$$

We unroll until  $k = \log_3(n) - 1$ .

The non recursive work each time we unroll is  $\frac{n^2}{3^i}$

Sum of non recursive work is  $\sum_{n=0}^{\log_3(n)-1} \frac{n^2 * 2^i}{3^i}$

$$\begin{aligned} T(n) &= \sum_{n=0}^{\log_3(n)-1} n^2 * \left(\frac{2}{3}\right)^i \\ &= n^2 * \frac{1 - (2/3)^{\log_3(n)}}{1 - (2/3)} \\ &= 3n^2 * (2/3)^{\log_3(n)/\log_3(2/3)(3)} \\ &= 3n^2 * n^{1/\log_3(2/3)(3)} \end{aligned} \quad \in \Theta(n^{1/\log_3(2/3)(3)})$$

□