CMSC351 Spring 2025 (§0101,0201,0301) Homework 4

Due Thursday Mar 6, 2025 by 23:59 on Gradescope.

Directions:

- Homework must be done on printouts of these sheets and then scanned properly, or via latex, or by downloading, writing on the PDF, and uploading.
- Do not use your own blank paper!
- The reason for this is that gradescope will be following this template to locate the answers to the problems so if your answers are organized differently they will not be recognized.
- Tagging is automatic, do not manually tag.

---

1. Here is the pseudocode for Binary Search with some code added:

```
function binarysearch(A,TARGET)
    L = 0
    R = n-1
    i = 0
    while L <= R
        i = i + 1
        C = floor((L+R)/2)
        if A[C] == TARGET
            print(i)
            return C
        elif TARGET < A[C]
            R = C-1
        elif TARGET > A[C]
            L = C+1
        end if
    end while
    print(i)
    return FAIL
end
```

(a) For the following list if we searched for each element in the list (separately) enter the value of `i` which will be printed right before the `return C`. [8 pts]

Index	0	1	2	3	4	5	6	7
Entry	2	5	7	8	10	13	17	20

`i`-value	3	2	3	1	3	2	3	4

(b) We search for a real number x which is not in the list and the `print` statement prints the `i`-value 4 right before the `return FAIL`. What can you say about the value of x? [4 pts]

Possible x-values	$17 < x < 20$ or $20 < x$

2. Here is the pseudocode for Binary Search: [20 pts]

```
function binarysearch(A,TARGET)
    L = 0
    R = n-1
    while L <= R
        C = floor((L+R)/2)
        if A[C] == TARGET
            return C
        elif TARGET < A[C]
            R = C-1
        elif TARGET > A[C]
            L = C+1
        end if
    end while
    return FAIL
end
```

Suppose Binary Search is applied to a list and the `TARGET` is `A[0]`. Prove by induction that for any $N \geq 0$ that if the list has length $3 \cdot 2^N - 2$ then the `while` loop passes exactly $N+1$ times.

Solution:

Base Case: When $N = 0$ the list has length $3 \cdot 2^0 - 2 = 1$ and so the `while` loop passes once since $L == R$ and that is exactly $N+1$ times.

Inductive Step: Suppose $N \geq 0$.

- We assume that for a list of length $3 \cdot 2^N - 2$ the `while` loop passes exactly $N+1$ times.
- We claim that for a list of length $3 \cdot 2^{N+1} - 2$ the `while` loop passes exactly $N+2$ times.

Suppose we have a list of length $3 \cdot 2^{N+1} - 2$. The `while` loop passes once and then since the list has even length it is the case that after the loop the length of the sublist we are looking at is:

$$\frac{3 \cdot 2^{N+1} - 2}{2} - 1 = 3 \cdot 2^N - 2$$

By the induction assumption the `while` loop passes $N+1$ times on this new sublist and so in total it passes $N+2$ times.

3. Consider the recurrence relation:

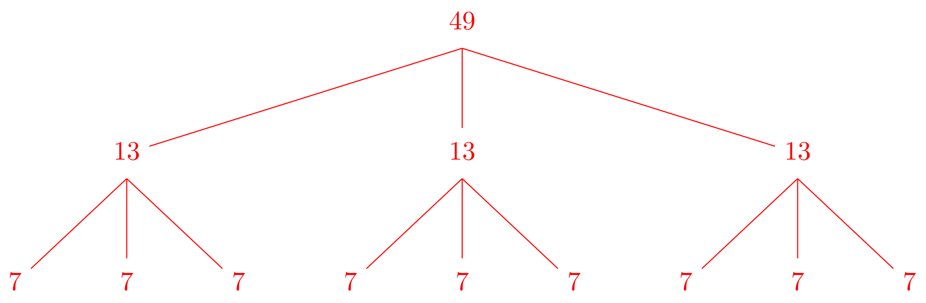
$$T(n) = 3T\left(\lfloor n/4 \rfloor\right) + 3n + 1 \quad \text{with} \quad T(0) = 2, T(1) = 7$$

(a) Calculate each of the following and simplify. [10 pts]

$T(3) =$	16
$T(4) =$	34
$T(12) =$	85

(b) Draw the tree corresponding to $T(16)$. Simplify all node values. [8 pts]

Solution:



4. Consider this recurrence relation: [15 pts]

$$T(n) = 2T(n-3) + 2 \quad \text{with } T(0) = T(1) = T(2) = 25$$

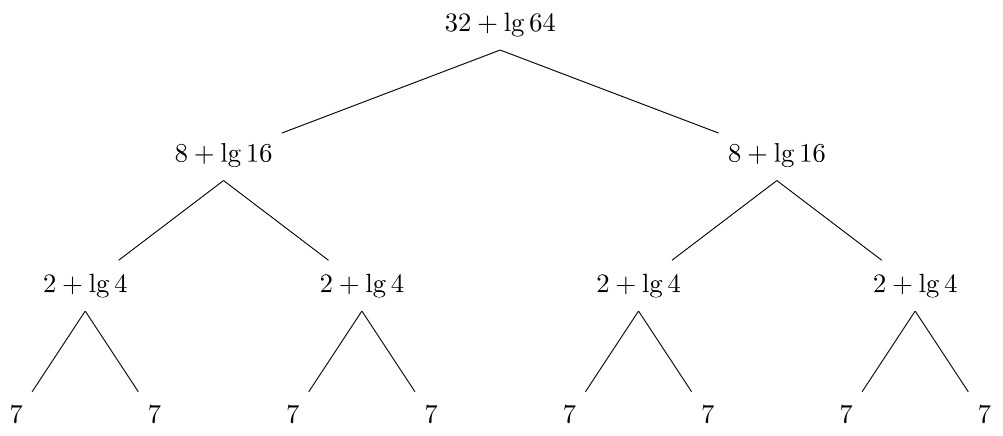
Use digging down to solve the recurrence relation specifically for when n is a multiple of 3. Simplify your answer as much as possible.

Solution:

We have:

$$\begin{aligned}
T(n) &= 2T(n-3) + 2 \\
&= 2(2T(n-6) + 2) + 2 = 2^2\, T(n-6) + 4 + 2 \\
&= 2^2(2T(n-9) + 2) + 4 + 2 = 2^3\, T(n-9) + 8 + 4 + 2 \\
&= \ldots \\
&= 2^{n/3}\, T(0) + 2 \sum_{i=0}^{n/3-1} 2^i \\
&= 25 \cdot 2^{n/3} + 2\left(2^{n/3} - 1\right) \\
&= 27 \cdot 2^{n/3} - 2
\end{aligned}$$

5. Here is a tree corresponding to $T(64)$ for an unknown recurrence relation: [10 pts]



Fill in the details for the corresponding recurrence relation:

$$T(n) = \boxed{2}\, T(n/4) + \boxed{\tfrac{n}{2} + \lg n}$$

$$T(1) = \boxed{7}$$

Put scratch work below. Scratch work is not graded but may be used for regrade partial credit.

6. For each of the following recurrence relations fill in the Θ time complexity given by the Master Theorem. If the Master Theorem does not apply write NA. [25 pts]

Relation	Θ of?
(a) $T(n) = 3T(n/2) + n^2$	n^2
(b) $T(n) = 4T(n/2) + n^2$	$n^2 \lg(n)$
(c) $T(n) = 2T(n/4) + n^{0.51}$	$n^{0.51}$
(d) $T(n) = 2^n T(n/2) + n^n$	NA
(e) $T(n) = 16T(n/4) + n$	n^2

Put scratch work below; Scratch work is not graded but may be used for regrade partial credit.