

Homework 19

Due Date March 23, 2023
Name **Blake Raphael**
Student ID **109752312**
Collaborators **Alex Barry, Brody Cyphers, and Ben Kohav**

Contents

1	Instructions	1
2	Honor Code (Make Sure to Virtually Sign)	2
3	Standard 19: Mergesort	3
3.1	Problem 1 (0.5 Points)	3
3.2	Problem 2 (1 Point)	4
3.3	Problem 3 (1 Point)	5
3.4	Problem 4 (1.5 Points)	6

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).

□

3 Standard 19: Mergesort

3.1 Problem 1 (0.5 Points)

Problem 1. Write down a recurrence relation that models the running time of Mergesort and give its asymptotic solution.

Answer.

$$T(n) = \begin{cases} \Theta(1) & : n \leq 1, \\ 2T(n/2) + \Theta(n) & : n > 1. \end{cases}$$

We find that we have $\frac{n}{2^i} \cdot 2^i$

We then find that $k = \log_2(n)$

We then find $T(n) = \sum_{i=0}^{\log_2(n)} n$

So, $T(n) = n \log_2(n)$ and $T(n) \in \Theta(n \log n)$

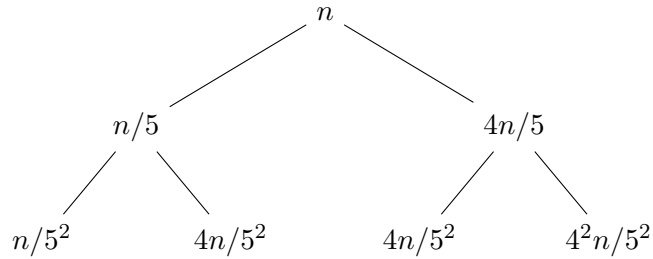
□

3.2 Problem 2 (1 Point)

Problem 2. Consider a modified Mergesort algorithm that at each recursion splits an array of size n into two subarrays of sizes $n/5$ and $4n/5$, respectively. Write down a recurrence relation for this modified Merge-Sort algorithm and give its asymptotic solution.

Answer.

$$T(n) = \begin{cases} \Theta(1) & : n \leq 1, \\ T(n/5) + T(4n/5) + \Theta(n) & : n > 1. \end{cases}$$



At level i , we see that:

$$\begin{aligned} \frac{4^i}{5} n &= 1 \\ \frac{4^i}{5} &= \frac{1}{n} \\ \log\left(\frac{4}{5}\right)^i &= \log\left(\frac{1}{n}\right) \\ x &= \log_2 n \\ i &= \log_{\frac{5}{4}} n \end{aligned}$$

So $T(n) \in \Theta(n \log n)$

□

3.3 Problem 3 (1 Point)

Problem 3. Consider a modified Mergesort algorithm that at each recursion splits an array of size n into two subarrays of sizes 2 and $n-2$, respectively. Write down a recurrence relation for this modified Merge-Sort algorithm and give its asymptotic solution.

Answer.

$$T(n) = \begin{cases} \Theta(1) & : n \leq 1, \\ T(2) + T(n-2) + \Theta(n) & : n > 1. \end{cases}$$

$$\begin{aligned} T(n) &= T(n-4) + \Theta(n-2) + \Theta(n) \\ &= T(n-6) + \Theta(n-4) + \Theta(n-2) + \Theta(n) \\ &= T(n-2k) + \Theta(n-(2k-2)) + \dots + \Theta(n) \end{aligned}$$

So $T(n) \in \Theta(n^2)$

□

3.4 Problem 4 (1.5 Points)

Problem 4. While we have focused on runtime analysis, we are also concerned with the amount of space an algorithm uses. Space analysis is very similar to time analysis in that we only care about dynamic data structures (e.g., list size n), we ignore literals (e.g., ints), and we report the total space used with respect to the input size (e.g., $O(n)$).

What is the space used by the Mergesort implementation below? (Hint: The total space an algorithm uses is the sum of the space used by all of its subproblems.)

```
def merge(A, B):
    a = 0
    b = 0
    R = []

    while a < len(A) or b < len(B):
        if a == len(A) or (b < len(B) and A[a] >= B[b]):
            R.append(B[b])
            b+=1
        else:
            R.append(A[a])
            a+=1

    return R

def mergesort(L):
    if len(L) <= 1:
        return L
    A = mergesort(L[:int(len(L)/2)])
    B = mergesort(L[int(len(L)/2):])
    return merge(A, B)
```

Answer. We find the space of Mergesort by finding the sum of the space used by each branch that is executed for the Mergesort lists. When we branch out on A, we find that it goes to $\frac{n}{2^k}$ layers. So we find that the space used

by the Mergesort implementation $T(n) = \sum_{i=0}^k \left(\frac{n}{2^i}\right)$ □