

Problem Set 7

Due Date TODO
Name John Blackburn
Student ID Jobl2177
Collaborators List Your Collaborators Here

Contents

Instructions	1
Honor Code (Make Sure to Virtually Sign the Honor Pledge)	2
1 Standard 19: Tree Method.	3
2 Standard 20: Quicksort.	4
2.1 Part (a)	4
2.2 Part (b)	5
2.3 Part (c)	6

Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Useful links and references on \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 \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 Honor Code). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign the Honor Pledge)

Problem HC. On my honor, my submission reflects the following:

- 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.

In the specified region below, clearly indicate that you have upheld the Honor Code. Then type your name.

(Agreed). I, John Blackburn, agree to adhere to the honor code

□

1 Standard 19: Tree Method.

Problem 1. 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**.

$$T(n) = \begin{cases} 1, & n < 3 \\ 3T(n/3) + n^2, & \text{otherwise.} \end{cases}$$

Answer. My tree will be composed of the non-recursive work done at each call, that being: n^2 . So starting with this as the source node of my tree we can determine the amount of work done at each level of the tree and find the pattern to discern a function $T(n)$ and $(f(n))$ such that $T(n) \in \Theta(f(n))$. Below embedded is my tree and work to discern the pattern the tree reveals:

Input	# of nodes	Tree	Row sum cost	depth
$T(n)$	1	n	$2^0 \left(\frac{n^2}{3^0} \right)$	0
$T\left(\frac{n}{3}\right)$	2	$\frac{n}{3} \quad \frac{n}{3}$	$2^1 \left(\frac{n^2}{3^1} \right)$	1
$T\left(\frac{n}{3^2}\right)$	4	$\frac{n}{3^2} \quad \frac{n}{3^2} \quad \frac{n}{3^2} \quad \frac{n}{3^2}$	$2^2 \left(\frac{n^2}{3^2} \right)$	2
$T\left(\frac{n}{3^3}\right)$	8	$\frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3} \quad \frac{n}{3^3}$	$2^3 \left(\frac{n^2}{3^3} \right)$	3
...
$T\left(\frac{n}{3^k}\right)$

1 Observe that the i -th level of the tree the non-recursive work done is $\left(\frac{n}{3^i}\right)^2$. Furthermore, there are 2^i nodes at level i . So the total non-recursive work done at level i is:

$$\left(\frac{n}{3^i}\right)^2 * 2^i = n^2 \left(\frac{2^i}{9^i}\right)$$

I now determine the number of levels in the tree. let k denote this. Now I solve for k , I hit a base case when:

$$\frac{n}{3^k} < 3$$

and solving for k we get:

$$k > \log_3(n) - 1$$

Thus, the total work done for the recurrence relation is:

$$T(n) = \sum_{i=0}^{\log_3(n)-1} n^2 \left(\frac{2}{9}\right)^i = n^2 \sum_{i=0}^{\log_3(n)-1} \left(\frac{2}{9}\right)^i = n^2 \left(\frac{1 - \left(\frac{2}{9}\right)^{\log_3(n)}}{1 - \frac{2}{9}} \right) = \frac{9n^2}{7} - \left(\frac{9}{7}\right) \left(\frac{2}{9}\right)^{\log_3(n)}$$

Thus, $T(n) \in \Theta(n^2)$

□

2 Standard 20: Quicksort.

2.1 Part (a)

Problem 2. (a) Write down a recurrence relation that models the **best case** running time of Quicksort, i.e. the case where PARTITION selects the **median** element at each iteration.

Answer.

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

□

2.2 Part (b)

- (b) Write down a recurrence relation that models the **worst case** running time of Quicksort, i.e. the case where PARTITION selects the **last** element at each iteration.

Answer.

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

□

2.3 Part (c)

- (c) Suppose that we modify $\text{PARTITION}(A, s, e)$ so that it chooses the median element of $A[s..e]$ in calls that occur in nodes of even depth of the recursion tree of a call $\text{QUICKSORT}(A[1, \dots, n], 1, n)$, and it chooses the maximum element of $A[s..e]$ in calls that occur in nodes of odd depth of this recursion tree.

Assume that the running time of this modified PARTITION is still $\Theta(n)$ on any subarray of length n . You may assume that the root of a recursion tree starts at level 0 (which is an even number), its children are at level 1, etc.

Your job is to write down a recurrence relation for the running time of this version of QUICKSORT given an array n distinct elements and solve it asymptotically, i.e. give your answer as $\Theta(f(n))$ for some function $f(n)$. Show your work.

Answer. For the odd depth, when the algo chooses the partition of the maximum element:

We already found that the recurrence relation for this is:

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

So to solve recursively we start at (I will replace $\Theta(n)$ with cn for clarity to find the pattern):

$$T(n) \leq T(n-1) + c(n)$$

$$T(n) \leq (T(n-2) + c(n-1)) + c(n)$$

$$T(n) \leq (T(n-3) + c(n-2) + c(n-1)) + c(n)$$

From here I can discern the pattern of the non-recursive work which is:

$$c \sum_{i=1}^n i = \left(\frac{cn(n-1)}{2} \right)$$

and from this we can see that:

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

As for the even depth, we have that:

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

Taken from part a.

Now to solve this asymptotically:

$$T(n) \leq T(n/2) + c(n)$$

$$T(n) \leq (T(n/4) + c(n/2)) + c(n)$$

$$T(n) \leq (T(n/8) + c(n/4) + c(n/2)) + c(n)$$

From here I can discern the pattern of the non-recursive work which is:

$$f(n) = c \sum_{i=0}^{\log(n)} n = cn \log(n)$$

Therefore:

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

So, overall we can determine that the function will run at $T(n) \in \Theta(n \log n)$ on the even depths of the recurrence and $T(n) \in \Theta(n^2)$ on the odd depths. To compute the total work done by the function, I find the sum of the non-recursive work done from the even depths and the odd depths. This is given by $\frac{cn(n-1)}{2} + cn \log(n)$. From which we take the higher order term and find that as a whole $T(n) \in \Theta(n^2)$.

□