CMSC351 Spring 2025 (§0101,§0201,§0301) Homework 6

Due Thursday Mar 27, 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. For each of the following lists your friend claims that the list was the result of running `partition` on some unknown list. For each, identify what the `pivotvalue` could have been. List all values or NONE if not possible. [8 pts]

`A`	Possible `pivotvalue`
`[7,6,5,8,10,9]`	
`[4,1,7,12,10,15,20,17]`	
`[1,2,3,4,5]`	
`[3,10,5,7,8,4,2]`	

Scratch Work; Scratch Work is Not Graded:

2. Consider the following pseudocode for `partition`: [12 pts]

```
function partition(A,L,R)
    pivotkey = A[R]
    t = L
    for i = L to R-1 inclusive:
        if A[i] <= pivotkey:
            A[t] <-> A[i]
            t = t + 1
        end if
    end for
    A[t] <-> A[R]
    return t
end function
```

Suppose we call `partition(A,0,5)` on `A=[10,1,9,2,8,5]`. Show the state of the list at the indicated instances.

Initial A	10	1	9	2	8	5
After $i = 0$ ends						
After $i = 1$ ends						
After $i = 2$ ends						
After $i = 3$ ends						
After $i = 4$ ends						
After final swap						

3. Consider the following pseudocode for `quicksort` with a `print` statement added: [14 pts]

```
function quicksort(A,L,R)
    if L < R:
        print(A)
        resultingpivotindex = partition(A,L,R)
        quicksort(A,L,resultingpivotindex-1)
        quicksort(A,resultingpivotindex+1,R)
    end if
end function
function partition(A,L,R)
    see previous problem
end function
```

Suppose we call `quicksort(A)` on `A=[33,22,92,47,62,44,89]`. The `print` statement runs exactly four times. What is printed each time?

First Time							
Second Time							
Third Time							
Fourth Time							

Scratch Work; Scratch Work is Not Graded:

4. There is an alternate version of Quick Sort which has two pivot values. Essentially we choose two pivot values $p_1 \leq p_2$ and the partition process partitions the list so that we have:

$$[\text{values} \leq p_1, p_1, p_1 \leq \text{values} \leq p_2, p_2, p_2 \leq \text{values}]$$

It is a fact that this partition process is $\Theta(n)$.

(a) In the best case the list would be divided into thirds. What would the corresponding recurrence relation be? [6 pts]

$T(n) =$	

(b) Solve this with the Master Theorem and put the details here: [10 pts]

Case	
a	
b	
c	
Result $T(n) = \Theta$ of what?	

5. In your own words explain why Merge Sort would be better than Quick Sort for lists which start reverse-sorted. [10 pts]

6. Here is a bit more mathematical detail behind the average case for Quick Sort. For the average case we determined that:

$$T(n) = \frac{1}{n} \sum_{k=0}^{n-1} [T(k) + T(n-k-1) + \Theta(n)]$$

For simplicity assume in fact that:

$$T(n) = \frac{1}{n} \sum_{k=0}^{n-1} [T(k) + T(n-k-1) + n]$$

(a) Prove that: [20 pts]

$$T(n) = \left[\frac{2}{n} \sum_{k=0}^{n-1} T(k) \right] + n$$

Solution:

(b) From the above it can be shown (don't - it's not obvious!) that for $n \geq 3$ we have: [20 pts]

$$\frac{T(n)}{n+1} \leq 2 \sum_{k=2}^{n} \frac{1}{k}$$

In addition, recall from Calculus 2 that:

$$\sum_{k=2}^{n} \frac{1}{k} \leq \int_{1}^{n} \frac{1}{x} \, dx$$

Use these two facts to prove that:

$$T(n) = \mathcal{O}(n \lg n)$$

Solution: