CMSC351 Spring 2025 (§0101,§0201,§0301) Homework 7

Due WEDNESDAY Apr 2, 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. Suppose we draw a decision tree for Bubble Sort applied to a list of length n.

   (a) What will the height of the tree be? Simplify your answer. [5 pts]

Height	$\frac{(n-1)(n)}{2}$

   (b) Explain. [10 pts]

   Solution:

   Our first pass makes $n-1$ comparisons, the second makes $n-2$, and so on until the last pass makes 1 comparison. In total then:

$$(n-1)+(n-2)+\ldots+(2)+(1)=\sum_{i=1}^{n-1} i=\frac{(n-1)(n)}{2}$$

2. Explain why for any sorting algorithm (not just comparison-based ones) which works by moving elements around that for any n there is at least one list which will take at least $C(n-1)$ time for some constant C to sort the list. [10 pts]

   Note: This does not need to be a formal proof but should be a concise explanation.

   Solution:

   For any n consider a list which is reverse-sorted. In order to sort this list we will need to move at least $n-1$ elements since if n is even we need to move all of them and if n is odd we need to move $n-1$ of them. Since each move takes at least constant time C we have our result.

3. What would be the (simplified) time complexity of using Counting Sort to sort a list of n integers ... [9 pts]

Restriction	Θ of what?
... between 0 and 7 inclusive?	n
... between 0 and $n \lg n$ inclusive?	$n \lg n$
... between n and n^3 inclusive?	n^3

4. We are using counting sort to sort the following array of numbers. The ? mean the values are unknown:

Index	0	1	2	3	4	5	6
Value	?	?	?	2	?	3	?

(a) Here is the helper array `POS` after the first (non-cumulative) step. Fill in the blanks: [6 pts]

Index	0	1	2	3
Value	2	3	1	1

(b) Here is the helper array `POS` after the second (cumulative) step. Fill in the blanks: [6 pts]

Index	0	1	2	3
Value	2	5	6	7

(c) Fill in the sorted array after Counting Sort has finished: [14 pts]

Index	0	1	2	3	4	5	6
Value	0	0	1	1	1	2	3

5. Suppose Radix Sort is used to sort the following list of strings. Show the state of the list after each iteration of the underlying sort. [10 pts]

Start	DIG	DIE	BID	DAD	BAD
After Iteration 1	BID	DAD	BAD	DIE	DIG
After Iteration 2	DAD	BAD	BID	DIE	DIG
After Iteration 3	BAD	BID	DAD	DIE	DIG

6. Suppose you made a mistake an wrote your Radix Sort code so that it sorted the leftmost digit first, then the next leftmost, and so on. Suppose your Broken Radix Sort is used with the following list of strings. Show the state of the list after each iteration of the underlying sort. [10 pts]

Start	824	126	844	240	840
After Iteration 1	126	240	824	844	840
After Iteration 2	126	824	240	844	840
After Iteration 3	240	840	824	844	126

7. Suppose Radix Sort (with underlying Counting Sort) is applied to a list of n^2 integers (in base 10) between 0 and $10^{\log_{10} n} - 1$ inclusive. For simplicity, you can assume that $\log_{10} n$ is an integer.

(a) What is the running time (in Θ notation) of Radix Sort in the above scenario? [4 pts]

Θ of the scenario described above?	$\Theta(n^2 \lg n)$

(b) Show your work for (a). [6 pts]

The numbers from 0 to $10^{\log_{10} n} - 1$ inclusive have $\log_{10} n$ digits. For example when $n = 1000$ this is stating that the numbers from 0 to $10^{\log_{10} 1000} = 999$ have $\log_{10} 1000 = 3$ digits. Thus, the running time of Radix Sort is $\Theta(\log_{10} n(n^2 + 9))$ or $\Theta(n^2 \lg n)$.

8. Suppose we run Radix Sort on the following list where the digit `x` is unknown: [10 pts]

Original List	`6x2`	`46x`	`108`	`8x8`	`28x`
After Iteration 1	`6x2`	`46x`	`28x`	`108`	`8x8`
After Iteration 2	`108`	`6x2`	`8x8`	`46x`	`28x`
After Iteration 3	`128`	`28x`	`46x`	`6x2`	`8x8`

What are the possible digit values for `x`? List them explicitly and in order.

Possible Values for `x` are:	$2, 3, 4, 5$

Scratch work; Not graded but may be used for regrade partial credit:

Iteration 1 tells us that $2 \leq x \leq 7$.
Iteration 2 tells us that $1 \leq x \leq 5$.
Iteration 2 tells us nothing else.