CS4040/5040
Homework # 3

due Wednesday, October 6, 2021
(CS4040 60 pts / CS5040 70 pts)

1. (5 pts.) Draw a decision tree that achieves the smallest average number of comparisons to sort a list of four items into the order $1, 2, 3, 4$. What is the height of your tree? What is the average number of comparisons needed for your tree assuming all orders are equally likely? What are the best case and worst cases for your tree? Does your tree model any of the standard sorts we have studied?

2. (5 pts.) Give a detailed description of an algorithm (pseudo code) that will sort n integers in the range 1 to n^3 in $O(n)$ time. Be sure to justify why your algorithm takes $O(n)$ time.

3. (6 pts.) For each of the following standard sorts, explain exactly why it is either a stable sort, or an unstable sort. Clearly state any assumptions you make.

   (a) quicksort
   (b) bubble sort
   (c) insertion sort
   (d) counting sort
   (e) heap sort
   (f) bucket sort
   (g) shell sort

4. (4 pts.) Show the steps in sorting the following English words using radix sort. Start with them in this exact relative order. Note, all these words are actual words in the scrabble dictionary!

   THIG PLOD BACK EVIL SCAG EDGE YEGG PLOT DOGE ACTS PLOW AWAY RANG BABE ACRE WING BABY PLUG ACTA BABU PLOP FILL ACNE DOPE BABA

5. (10 pts.) You are given an array of integers, each with an unknown number of digits. You are also told the total number of digits of all the integers in the array is n. Provide an algorithm that will sort the array in $O(n)$ time no matter how the digits are distributed among the elements in the array. (e.g. there might be one element with n digits, or $n/2$ elements with 2 digits, or the elements might be of all different lengths, etc. Be sure to justify in detail the run time of your algorithm.

6. (10 pts.) Problem 7-4, page 188.

7. (10 pts.) To find the top wage earner in the country (i.e. the one with the highest reported IRS income) requires $O(n)$ time (use the algorithm that finds the maximum). Suppose instead we want to find the set of people that are in the top $k\%$ by reported income (not necessarily in sorted order). Describe an algorithm (psuedocode) to solve this problem in the smallest possible asymptotic time complexity. Argue that your algorithm is optimal. Analyze the space and time complexity of your algorithm in detail.

8. (10 pts.) In the Select algorithm the input elements are divided into groups of 5 elements. Be sure to justify your answers to the questions below.

   (a) Does the algorithm run in linear time if the elements are divided into groups of 3 elements?

   (b) Does the algorithm run in linear time if the elements are divided into groups of 7 elements?

   (c) **CS5040 only:** What is the optimal number of elements to divide the input into if we care only about the detailed time complexity of the algorithm. Note: I am NOT talking about the big-O complexity here, but rather the exact number of operations needed to calculate the answer. Show a detailed analysis for why you think it is the optimum number of elements.