COSC-311 Sample Midterm Questions

Note: There will be four problems on the actual midterm. The problems on this handout are meant to give you a sense of the types of questions I might ask. This study guide is not comprehensive: there are topics we have covered in class that are not represented in the sample problems. Everything we have done in class or on homework is fair game for the midterm.

1 Runtime Analysis

(a) What does it mean for $f(n)$ to be big-O of $g(n)$?

(b) Here is a recurrence:

$$T(n) = 2T(n/3) + 4n$$

Prove by induction that $T(n) = O(n)$.

2 Stooge Sort

Consider the following sorting algorithm:

```
Stoogesort(A, i, j){
   if A[i] > A[j]
      swap(A[i], A[j])
   if (j - i + 1) > 2
         t = (j - i + 1) / 3
         Stoogesort(A, i, j-t)
         Stoogesort(A, i+t, j)
         Stoogesort(A, i, j-t)
   return A
}
```

(a) Write a recurrence for the runtime of this algorithm.

(b) Is Stooge Sort more or less efficient than Insertion Sort? Why? You may want to use the Master Theorem, draw a recursion tree, or use some other strategy to reason about the asymptotic runtime of Stooge Sort.

3 Sorting Runtimes

(a) Fill in each (non-gray) entry of the table to give the runtime of each algorithm in terms of n, the number of elements in the input array. Be as specific as possible (i.e., give Θ bounds wherever known).

Algorithm	**Worst-case runtime**	**Average-case runtime**
Mergesort		
Insertion sort		
Heapsort		
Deterministic Quicksort		
Selection sort		

(b) An array A is said to be k-sorted if it has the property that for every element i in array A, i's position in array A is at most k away from i's position in a sorted array of the same data. In terms of k and n, what is the asymptotic runtime of insertion sort on k-sorted data? Explain why in a sentence or two.

4 Recurrences

Allie and Brandon each have come up with a divide-and-conquer algorithm to solve a problem.

- Allie's algorithm (algorithm A) splits a problem of size n into four pieces, each of size $n/2$, solves the pieces recursively, and takes time $30n$ to combine the results.
- Brandon's algorithm (algorithm B) takes time n^3 to turn a problem of size n into a smaller problem of size $n/2$, which it then solves recursively.

(a) Write a recurrence for the runtime of Allie's algorithm, $T_A(n)$, and a recurrence for the runtime of Brandon's algorithm, $T_B(n)$.

(b) Which algorithm has an asymptotically better runtime? Justify your answer by solving each recurrence using the Master Theorem.

5 Sorting Instability

(a) What does it mean for a sorting algorithm to be stable?

(b) What is an example of a sorting algorithm that is not inherently stable? Explain why it isn't stable.

(c) Suppose you wanted to adapt the unstable sorting algorithm from part (b) so that it is stable, and assume you are sorting integers. How could you modify the information stored in the array (i.e., modify the data you are sorting) so that the algorithm is guaranteed to be stable?

6 A Sorting Mishap

Suppose that you're in the middle of sorting an array when one of the elements in the array is randomly changed to a completely different value. There are several possibilities for what could happen to the final output:
(1) The final array will be mostly sorted, with one or two elements out of place
(2) The final array will have two separate sorted sections
(3) The final array will not even resemble a sorted array

For each of the following algorithms, which option (1, 2, or 3) is the *worst* possible outcome? Explain why in a sentence or two.

(a) Insertion sort

(b) Mergesort

(c) Selection sort