Midterm Study Guide Computer Science 311 Fall 2019 Gardner/Rager

1. You should know the definitions of O, Θ, Ω

2. **The Master Theorem**

You should be able to use the Master Theorem. We will put the statement of the theorem on the test sheet if we ask about it. You do NOT have to memorize the theorem. The statement of the theorem is:

Theorem 4.1 (Master Theorem) Let $a \geq 1$ and $b > 1$ be constants, let f(n) be an asymptotically positive function. Let T(n) be defined by the recurrence:

$T(n) = aT(n/b) + f(n)$ for $n > 0$ Then:

1. If $f(n) = O(n^{(\log_b a - \varepsilon)})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{(\log_b a)})$
2. If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \lg n)$
3. If $f(n) = \Omega(n^{(\log_b a + \varepsilon)})$ for some constant $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = O(f(n))$.

2. You should be able to hand simulate and describe the running times of all of the following. (Describe the running time means give a convincing albeit informal argument, not merely state. Note that we did not discuss all of these runtimes in class, but in general you should be comfortable reasoning about the running time of a given piece of pseudocode.):

a. Insertion Sort
b. Selection Sort
c. Merge Sort
d. Quick Sort
e. Heap Sort
f. Interval Scheduling (Greedy Algorithm)
g. Multiplication of n-digit numbers (Divide and Conquer)
h. Dijkstra's SSSP

3. You should understand the two major algorithm paradigms, greedy and divide-and-conquer.
    a. You should be able to give examples of these algorithms
    b. Given a novel problem, you should be able to devise an algorithm. We will tell you whether you should use a greedy algorithm or a divide-and-conquer algorithm.

4. You should understand some of the basic properties of sorts – stability and in-place. You should know which sorts are stable and how much extra memory is needed for each of the sorts.