COSC 311: ALGORITHMS
MINI 4 SOLUTIONS

1. Applying the Master Theorem.

(a) Let $T(n) = 4T(n/4) + \sqrt{n}$. Use the Master Theorem to determine the Θ class of $T(n)$.

Solution: Here we have $a = 4$, $b = 4$, and $f(n) = \sqrt{n}$.
First consider $n^{\log_b(a)} = n^{\log_4(4)} = n^1 = n$.
We see that for $\epsilon < 1/2$, we have $f(n) = \sqrt{n} = O(n^{\log_b(a)-\epsilon}) = O(n^{1-\epsilon})$.
This indicates that we are in the first case, and therefore we know that $T(n) = \Theta(n^{\log_b(a)}) = \Theta(n)$.

(b) Let $T(n) = 9T(n/3) + 2n^2$. Use the Master Theorem to determine the Θ class of $T(n)$.

Solution: Here we have $a = 9$, $b = 3$, and $f(n) = 2n^2$.
First consider $n^{\log_b(a)} = n^{\log_3(9)} = n^2$.
Thus we see that we have $f(n) = 2n^2 = \Theta(n^{\log_b(a)}) = \Theta(n^2)$.
This indicates that we are in the second case, and therefore we know that $T(n) = \Theta(n^{\log_b(a)} \log(n)) = \Theta(n^2 \log(n))$.

2. Interpreting the Master Theorem Why do we compare $n^{log_b a}$ and $f(n)$ when deciding which case of the Master Theorem applies? What does this comparison tell us about where most of the work happens in the algorithm were studying?

Solution: Comparing $n^{log_b a}$ and $f(n)$ allow us to essentially compare the work we do at the root versus at the leaves. That is, if $f(n) = O(n^{log_b a})$ then we know that the work we do at each step (at the root) is less than the work we do moving through the steps of the algorithm (at the leaves). If $f(n) = \Omega(n^{log_b a})$ then we know that the work we do at each step (at the root) is greater than the work we do moving through the steps of the algorithm (at the leaves). If $f(n) = \Theta(n^{log_b a})$ then the two are roughly equal, so the overall runtime is equal to the work per level multiplied by the number of levels in the recursion tree. After making this comparison, we take the larger of the two ($n^{log_b a}$ or $f(n)$) and weight that more heavily in our assessment of the overall runtime of the algorithm, essentially setting $T(n) = max(\Theta(n^{log_b a}), \Theta(f(n)))$, or if the two are equal then setting $T(n) = \Theta(n^{log_b a} \log n)$.