HW 1.

1. Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

(a) $n^2$
(b) $n^3$
(c) $100n^2$
(d) $n\log n$
(e) $2^n$
(f) $2^{2n}$

2. Suppose we are comparing two different sorting algorithms on the same machine. For an input of size $n$, Algorithm A runs in $8n^2$ steps, while algorithm B runs in $64n\log n$ steps. For what value of $n$ does algorithm A beat algorithm B?

3. $f(n) = 10^6n^2$ and $g(n) = n^2$. Using the definition, show that $f(n) \in O(n^2)$.

4. Consider the following pseudo-code for Bubble sort

```java
for(int x=0; x<n; x++)
{
    for(int y=0; y<n-1; y++)
    {
        if(array[y]>array[y+1])
        {
            int temp = array[y+1];
            array[y+1] = array[y];
            array[y] = temp;
        }
    }
}
```
What is the Worst-case and Average-case running time for this code? Rewrite this code without adding any loops so that the best-case running time is now $\Theta(n)$. Also mention what input meets the best case condition.

5. Consider sorting $n$ numbers stored in an array $A$ by first finding the smallest element of $A$ and exchanging it with the element in $A[0]$. Then find the second smallest element of $A$, and exchange it with $A[1]$. Continue in this manner for the first $n - 1$ elements of $A$.

Write pseudocode for this algorithm, which is known as selection sort. Give the best case and worst case times of selection sort in $\Theta-$ Notation.