Jonathan Petersen
A01236750

CS2420 - Assignment 1

1. a. $f(n) = O(g(n))$
   b. $f(n) = \Theta(g(n))$
   c. $f(n) = O(g(n))$
   d. $f(n) = O(g(n))$
   e. $f(n) = \Omega(g(n))$

2. a. $O(n)$
   b. $O(n^2)$
   c. $O(n^3)$
   d. $O(n^2)$
   e. $O(n^4)$
   f. $O(n^2)$

3. Base Case:

   Assume n = 1. Solving for the sum from i = 1 to 1 of (2i - 1), we see that $2(1) - 1 = 1$. Solving for n^2, we find that $(1)^2 = 1$. Since 1 = 1, this equation is valid when n = 1.

   Inductive Step:

   Assume the equation is true for n - 1. Observe that the summation from i = 1 to n of (2i - 1) is equal to the summation from i = 1 to (n - 1) of (2i - 1) + (2n - 1), therefore, based on our assumption, this is also equal to $(n - 1)^2 + (2n - 1)$. By solving $(n - 1)^2 + (2n - 1) = n^2$, we find that indeed $(n - 1)^2 + (2n - 1) = n^2 - 2n + 1 + 2n - 1 = n^2$. Therefore, this equation is true for n = 1 and n > 1. ∎

4. a.

```
// Start with array array[] of size n

int minIndex = 0;
for(int i = 0; i < n - 1; i++) {
      for (int j = i + 1; j < n; j++) {
            if( array[j] < array[minIndex]) {
                  minIndex = j;
            }
      }
      std::swap(array[i], array[minIndex]);
}
```

   b. Worst Case running time: $O(n^2)$
   c. Best Case running time: $O(n^2)$

5. See Canvas Submission

6. See Canvas Submission