CSC 505 – HW 1
Greg Timmons

1.) Pseudo code for the selection sort algorithm follows :

```
for ( i from 1 to n-1 ) {
    low = i;
    for ( j from i to n ) {
        if ( arr[low] < arr[j] ){
            low = j;
        }
    }
    if( low != i ){
        swap( arr[low], arr[j] );
    }
}
```

In the above code the total count of operations:

$$\sum_{i=1}^{n-1}\sum_{k=i+1}^{n} 1 = \sum_{i=1}^{n-1} n - i = \left(\sum_{i=1}^{n-1} n\right) - \left(\sum_{i=1}^{n-1} i\right)$$
$$= n(n-1) - \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$$
$$\equiv \Theta(n^2)$$

All cases have the same runtime since there is not early exit to either loop, so the runtime is constant regardless of input

The following pseudo code could exhibit a better runtime; it contains extra logic to increment the outer loop when the value at i+1 is equal to the value at i since we know that there will be nothing lower than the value at i.

```
for ( i from 1 to n-1 ) {
    low = i;
    if( arr[i] == arr[i+1] ){
        continue;
    }
    for ( j from i to n ) {

        if ( arr[low] < arr[j] ){
            low = j;
        }
    }
    if( low != i ){
        swap( arr[low], arr[j] );
    }
}
```

In the case that all of the values in the input are the same, this would exhibit a best case runtime since the inner loop would never execute. The runtime becomes simply the outer loop which is trivial to see has a runtime of $\Theta(n)$.