William Porter
Assignment 1

1 a. $O(n)$ because the for loop has a worst case time complexity of $O(n)$, worst case the item is at the end of the loop. If the array is length 'n', the loop will run 'n' times, making it $O(n)$

b.

i: 1, 2, 3, 4, ..., n

(per i) j: 2, 3, 4, ..., $\sqrt{n}$

$O(n\sqrt{n})$

n $\log n$ $n\log n$ $\log n \log n$

but also another for loop with $O(n)$

so $O(n\sqrt{n} + n)$

since $n\sqrt{n}$ has a higher degree than n,

the time complexity is $\boxed{O(n^{3/2})}$

2. a since $6n^2 + 13n - 6 \leq cn^4$
$O(n^4)$ works, however n^2 would be a much better estimate

b. since $n^2 - 2n + 4 \geq n$
then $n^2 - 2n + 4 = \Omega(n)$

3 a. $O(2n^3 + 6n + 25n) = O(n^3)$

b. $O(\log_2 n^2 + (\log_2 n)^2 + 1n n + 21) = O(2\log n + \log n \cdot \log n + \log$
$= O(\log n \cdot \log n)$

4. the code will not work because the base case is never met. To make it work:

```
int fib(int n){
    if (n<=1){
        return n;
    return fib(n-1) + fib(n-2),
```

iterative Function:

(modeling 1, 1, 2, 3, 5, 8, 13, ...)

```
int fib (int n){
    if (n == 2){
        return 1;
    } else if (n == 1){
        return 1;
    }
    int num = 0;
    int first = 1;
    int second = 1;
    for (int i = 0; i < n-2; i++){
        int temp = second;
        second = second + first;
        first = temp;
    }
    num = second
    return num;
```

if n≤0 throw error