Name: __Edward Yaroslavsky______________ Date: ______9/18/19___

I pledge my honor that I have abided by the Stevens Honor System.

Point values are assigned for each question. Points earned: ____ / 100, = ____ %

1. Find an upper bound for $f(n) = n^4 + 10n^2 + 5$. Write your answer here: __O(n^4)_____ (4 points)

   Prove your answer by giving values for the constants c and n_0. Choose the smallest integral value possible for c. (4 points)
   c = 2
   n_0 = 4

2. Find an asymptotically tight bound for $f(n) = 3n^3 - 2n$. Write your answer here: __θ(n^3)_____ (4 points)

   Prove your answer by giving values for the constants c_1, c_2, and n_0. Choose the tightest integral values possible for c_1 and c_2. (6 points)
   c_1 = 2
   c_2 = 3
   n_0 = 2

3. Is $3n - 4 \in \Omega(n^2)$? Circle your answer: yes / no (2 points)

   If yes, prove your answer by giving values for the constants c and n_0. Choose the smallest integral value possible for c. If no, derive a contradiction. (4 points)
   $cn^2 \leq 3n - 4 \leq 3n$
   $cn^2 \leq 3n$
   $c \leq 3/n$
   $n \leq 3/c$
   This is a contradiction since n^2 is not a lower bound for $n \leq 3/c$.

4. Write the following asymptotic efficiency classes in **increasing** order of magnitude.
   $O(n^2)$, $O(2^n)$, $O(1)$, $O(n \lg n)$, $O(n)$, $O(n!)$, $O(n^3)$, $O(\lg n)$, $O(n^n)$, $O(n^2 \lg n)$ (2 points each)

   O(1), O(lg n), O(n), O(n lg n), O(n^2), O(n^2 lg n), O(n^3), O(2^n), O(n!), O(n^n)

5. Determine the largest size *n* of a problem that can be solved in time *t*, assuming that the algorithm takes *f(n)* milliseconds. *n* must be an integer. (2 points each)

   a. *f(n)* = n, *t* = 1 second __1000_______

   b. *f(n)* = $n \lg n$, *t* = 1 hour ___204094_______

c. *f(n)* = n^2, *t* = 1 hour ___1897_______

d. *f(n)* = n^3, *t* = 1 day ___442_______

e. *f(n)* = $n!$, *t* = 1 minute ____8______

6. Suppose we are comparing two sorting algorithms and that for all inputs of size n the first algorithm runs in $4n^3$ seconds, while the second algorithm runs in $64n\ lg\ n$ seconds. For which integral values of n does the first algorithm beat the second algorithm? ____2 through 6____________ (4 points) Explain how you got your answer or paste code that solves the problem (2 point):

   I graphed the two functions and observed the intersection points. From there I checked for the range where $4n^3$ was more efficient and took less time to complete. This was the range between 2 and 6, inclusive.

7. Give the complexity of the following methods. Choose the most appropriate notation from among O, Θ, and Ω. (8 points each)

```
int function1(int n) {
    int count = 0;
    for (int i = n / 2; i <= n; i++) {
        for (int j = 1; j <= n; j *= 2) {
            count++;
        }
    }
    return count;
}
```

Answer: ___ $\Theta(n * \log_2 n)$______

```
int function2(int n) {
    int count = 0;
    for (int i = 1; i * i * i <= n; i++) {
        count++;
    }
    return count;
}
```

Answer: __ $\Theta(n^{1/3})$_______

```
int function3(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            for (int k = 1; k <= n; k++) {
                count++;
            }
        }
    }
    return count;
}
```

Answer: ___ $\Theta(n^3)$______

```
int function4(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            count++;
            break;
        }
    }
    return count;
}
```

Answer: ___ $\Theta(n)$______

```
int function5(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        count++;
    }
    for (int j = 1; j <= n; j++) {
        count++;
    }
    return count;
}
```

Answer: ___ $\Theta(n)$______