Name: _Julia Nelson_________________ Date: ____9/18/19________

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: _$n^4 + 10n^2 + 5 <= c(n^4)$ for c=4 , n=2_ (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)

   $n^4 + 10n^2 + 5 <= c(n^4)$

   When c = 1 n =1 ---> 16 <= 1 NO

   When c = 2 n =2 ---> 61 <= 32 NO

   …

   When c = 4 n =2 ---> 16 <= 32 YES

   The smallest value of c to work is c = 4, and n = 2

2. Find an asymptotically tight bound for $f(n) = 3n^3 - 2n$. Write answer here: $C_2(n^2)<=2n^2-n<=C_1$ (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)

   $n_o >= 1, C_1 = 2, C_2 = 1$

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)

   When f(n) = n …… n $\Omega(n^2)$ ➔ $n >= n^2$

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)

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

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 ___10^3___

   b. *f(n)* = $n \lg n$, *t* = 1 hour _$2.04 * 10^5$__

c. $f(n) = n^2$, t = 1 hour ___1.8974 * 10^3___

d. $f(n) = n^3$, t = 1 day ___4.421 * 10^2___

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? ___n = 2, 3, 4___ (4 points)
Explain how you got your answer or paste code that solves the problem (2 point):
$4n^3 < 64 \lg n$ ➔ $n^2/16 < \lg n$
When…
n = 1 … 1/16 < 0 … NO
n = 2 … 0.25 < 0 … YES
n = 3 … 0.5625 < 1.584 … YES
n = 4 … 1 < 2 … YES
n = 5 … 1.5625 < 0.698 … NO

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: ___O(nlogn)___

```
int function2(int n) {
    int count = 0;
    for (int i = 1; i * i * i <= n; i++) {
        count++;
    }
    return count;
}
```
Answer: ___$\Theta(\sqrt[3]{n})$___

```
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: ___$O(n)$___