Name: Himanshu Rana Date: 9/24/18

"I pledge my honor that I have abided by the Stevens Honor System" – Himanshu Rana (hrana2)

Point values are assigned for each question. Points earned: ____ / 42, = ____ %

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

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

2. Find an asymptotically tight bound for $f(n) = 2n^2 - n$. Write your answer here: Θ (2 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. (3 points)
   c_1 = 1, c_2 = 2, n_0= 1

3. Is $3n - 4 \epsilon \Omega(n^2)$? Circle your answer: yes / **no**. (1 point)

   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. (2 points)

$$0 \leq cn^2 \leq 3n - 4 \leq 3n - 4n \ \forall n \leq 1$$
$$cn^2 \leq -n$$
$$cn^2 + n \leq 0$$
$$n(cn + 1) \leq 0$$
$$n \leq 0\,, cn + 1 \leq 0$$
$$n \leq -\frac{1}{c}$$

   This is a contradiction because it states that n has to be negative but the definition is that Ω(g(n)) ∃ a positive constant and n_0.

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

   a. *f(n)* = *n*, *t* = 1 second **1000**

b. *f(n)* = $n \lg n$, *t* = 1 hour **204,094**

c. *f(n)* = n^2, *t* = 1 hour $\sqrt{\mathbf{3600000}} \sim$ **1897**

d. *f(n)* = n^3, *t* = 1 day $\sqrt[3]{\mathbf{86400000}} \sim$ **442**
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? $\mathbf{2 \leq n \leq 6}$ (2 points)
   Explain how you got your answer or paste code that solves the problem (1 point):
   By graphing the two functions, I was able to see where they intersect and for what values the first function was lower than the second.

7. Give the complexity of the following methods. Choose the most appropriate notation from among O, Θ, and Ω. (3 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: $\Theta(\mathrm{nlgn})$

```
int function2(int n) {
    int count = 0;
    for (int i = 1; i * i * i <= n; i++) {
        count++;
    }
    return count;
}
```
Answer: Θ(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: $\Theta(\mathrm{n})$