CS 385, Homework 1: Analysis of Algorithms

Name: __**Katie Prescott**______________________ Date: ____**2/13/16**________

I pledge my honor that I have abided by the Stevens honor system.

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

1) Use the definitions of , Θ, and Ω to determine whether the following assertions are true or false. If true, give values for the appropriate constants. If false, explain the contradiction. (3 pts. each)

a) $n(n+1)/2 \in O(n^3)$ _**True:** $\boldsymbol{c = 1, n_0 = 0}$ ___________________________

b) $n(n+1)/2 \in O(n^2)$ _**True:** $\boldsymbol{c = 1, n_0 = 0}$ ___________________________

c) $n(n+1)/2 \in \Theta(n^3)$ __**False:** $\boldsymbol{n \leq \frac{1}{c_1}}$**, can't have n less than or equal to a constant.** ________

d) $n(n+1)/2 \in \Omega(n^3)$ __**False:** $\boldsymbol{n \leq \frac{1}{c_1}}$**, can't have n less than or equal to a constant.** ________

2) 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 pt. each)

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

3) Determine the largest size *n* of a problem that can be solved in time *t*, assuming that the algorithm takes *f(n)* milliseconds. (1 pt. each)

a) $f(n) = n, t = 1\ second$ ____$\boldsymbol{n = 1{,}000}$_____

b) $f(n) = n \lg n, t = 1\ hour$ ____$\boldsymbol{n \log_2 n = 3.6 \times 10^6}$_____

c) $f(n) = n^2, t = 1\ hour$ ____$\boldsymbol{n = \sqrt{3.6 \times 10^6}}$_____

d) $f(n) = n^3, t = 1\ day$ ____$\boldsymbol{n = \sqrt[3]{8.64 \times 10^7}}$_____

e) $f(n) = n!, t = 1\ minute$ ____$\boldsymbol{n! = 60{,}000}$_____

4) (3 pts.) Suppose we are comparing two sorting algorithms and that for all inputs of size n, the first algorithm runs in $8n^2$ seconds, while the second algorithm runs in $48n \log_2 n$ seconds. For which integral values of n does the first algorithm beat the second algorithm? ____$\boldsymbol{n \geq 1.141}$___________

Explain how you got your answer: **Based on the graphs, you can tell where the first function beats the second function by checking where they intersect, (where** $\boldsymbol{8n^2 = 48n \log_2 n}$**) which is** $\boldsymbol{n/(log_2 n) = 6}$ **and** $\boldsymbol{n = 1.141}$ **at that point. The first function beats the second function where** $\boldsymbol{n \geq 1.141}$ **because that is where** $\boldsymbol{8n^2}$ **is less than** $\boldsymbol{48n \log_2 n}$**.** ____________________________

5) Give the complexity of the following methods. Choose the most appropriate notation from among $O, \Theta, and\ \Omega$. (3 pts. 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: ____ $\boldsymbol{\Theta(n \lg n)}$ _____

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

Answer: ____ $\boldsymbol{\Theta(\sqrt{n})}$ _______

```
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: ____ $\boldsymbol{\Theta(n^3)}$_______

```
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: ____ $\boldsymbol{\Theta(n)}$ _______