Algorithms CS315 **Quiz 3** **Spring 2018**

Name: ____

1. (5 points) What does it mean that the worst case run time of an algorithm is in $\Theta(n^2)$?

   Answer: The maximum value of $T(n)$ for each n is greater than or equal to some n_0 is tightly bound by the function n^2.

2. (5 points) Consider the recursive function:

```
// Returns an integer
SomeRecursiveFunction(n)
   if n==1 then
     return 1
   else
     a = SomeRecursiveFunction(Floor(n/2))
     b = SomeRecursiveFunction(Ceiling(n/2))
     c = SomeRecursiveFunction(Floor(n/2))
     return (a+b+c)
```

   Give a tight asymptotic bound for $T(n)$ the runtime of `SomeRecursiveFunction` i.e. show $T(n) = \Theta(g(n))$ for some function $g(n)$.

   Answer: The runtime function for this function is $T(n) = 3T(n/2) + 1$. Using the master theorem Case 1 we obtain that $T(n) = \Theta(n^{\log_2 3})$.

3. (5 points) Find a tight asymptotic bound using the Master Method for the recurrence

$$T(n) = 4T(n/3) + n^2.$$

   Answer: We see that $n^2 = \Omega(n^{\log_3 4+\epsilon})$ for $\epsilon = 2 - \log_3 4$. Hence we are in Case 3 of the Master Method. So show regularity: $4(n/3)^2 = 4/9n^2$. So for $c_1 = 4/9$ and $n \geq 1$ the regularity condition holds. Therefore $T(n) = \Theta(n^2)$.

4. (5 points) Use the substitution method to find a tight asymptotic bound for

$$T(n) = 2T(\lfloor n/2 \rfloor) + n.$$

   (*Hint:* Find the bound by using the Master Theorem so you know what to guess.)

Answer: By the Master Method Case 2 we see that $T(n) = \Theta(n \log n)$. Suppose $T(n/2) \leq c(n/2)\log(n/2)$. We first show that $T(n) \leq cn \log n$:

$$\begin{aligned}
T(n) &= 2T(\lfloor n/2 \rfloor) + n \\
&\leq 2T(n/2) + n \\
&\leq c(n/2)\log(n/2) + n \\
&\leq c(n/2)\log n - c(n/2)\log 2 + n \\
&\leq cn/2 \log n - c(n/2) + n \\
&\leq cn \log n
\end{aligned}$$

for $c = 2$. A similar calculation shows that $0 \leq cn \log n \leq 2T(\lfloor n/2 \rfloor) + n$. Hence $T(n) = \Theta(n \log n)$.