CS 3343 (Spring 2018) Assignment 2 (100 points)

Due: Friday, Feb 2 before class starts

1. (40 points) Analysis of recursive algorithm. Consider the pseudocode of the following three algorithms for computing $2^n$, where $n$ is a non-negative integer.

```
Alg1 (n)
    if (n == 0) return 1;
    return 2 * Alg1(n - 1);
end

Alg2 (n)
    if (n == 0) return 1;
    return Alg2(n - 1) + Alg2(n - 1);
end

Alg3 (n)
    if (n == 0) return 1;
    m = floor (n / 2);
    p = Alg3(m);
    p = p * p;
    if (n % 2 == 1) // n is an odd number
        return 2 * p;
    else // n is an even number;
        return p;
    end
end
```

a. (6 points) Trace the three algorithms using two small examples ($n = 2$ and $n = 3$) to find out the outputs of the three algorithms, respectively.

b. (4 points) Prove the correctness of Alg1 using induction.

c. (12 points) Let $A(n)$, $B(n)$, and $C(n)$ be the running time of the three algorithms, respectively, as a function of $n$. Write down the recurrences for $A(n)$, $B(n)$, and $C(n)$.

d. (18 points) Analyze and compare the running time of the three algorithms above. You can use either recursion tree or master method to solve the recurrences.

2. (60 points) Assume that $T(1) \in \Theta(1)$. Solve the following recurrences using the recursion tree method.

   a. $T(n) = 4T(n/2) + n^2$
   b. $T(n) = 2T(n/2) + n^2$
   c. $T(n) = T(n - 2) + n$
   d. $T(n) = 4T(n/2) + n$
   e. $T(n) = 2T(n - 2) + 1$
   f. $T(n) = T(n/2) + T(n/3) + n$