COMP 4030/6030 Spring 2018. Assignment 6
Due: March 29, 2018 (before class starts)

1. (10 points) Use the Master's theorem to find the complexity (in terms of Theta) of this function $T(n) = n^2 + 4T(n/2)$.

2. (10 points) Use the Master's theorem to find the complexity (in terms of Theta) of this function $T(n) = n^2 + 6T(n/2)$.

3. (10 points) Use the Master's theorem to find the complexity (in terms of Theta) of this function $T(n) = n^2 + 3T(n/2)$.

4. (20 points) Find the running time equation of this Python function

```
def  foo(L):   # L is a list with n numbers
    if L==[]:
            return 5
    s = 0
    for x in L:
            s = s + x
    A = L[0: len(L)//2]
    return foo(A) + s
```

5. (10 points) Use the Master's theorem to find the complexity in terms of Theta of the running time equation in problem 4.

6. (20 points) Find the running time equation of this Python function

```
def  foo(L):   # L is a list with n numbers
    if L==[]:
            return 5
    s = 0
    for x in L:
            s = s + x
    A = L[0: len(L)//4]
    B = L[len(L)//4 : len(L)//2]
    C = L[len(L)//2 : 3*len(L)//4]
    return foo(A) + foo(B) + foo(C) + s
```

7. (10 points) Use the Master's theorem to find the complexity in terms of Theta of the running time equation in problem 6.

8. (10 points) Use a global table (dictionary) to cache the following function

```
def foo(n):
    if n==0:
            return 0
    if n==1:
            return 1
    if n==2:
            return 2
    if n%3 == 0:
            return foo(n-1) + foo(n-2) + foo(n-3)
    if n%3 == 1:
            return foo(n-1)
    return foo(n-1) + foo(n-3)
```

Plagiarism Policy:
You can discuss how to solve the problems with your classmates, but the solution must be your own. Using other people's solution will result in a zero for the assignment and possible additional penalties.

Submission:
Put your name as part of the file name and upload your submission to eCourseware Dropbox.