Homework Set #3

Assigned: Tuesday, 2/24  Due: Tuesday, 3/3 (start of lecture)

This assignment covers textbook material in Chapters 1-4.
Note: Refer to course website for homework policies.
Remember to attach signed honor statement.

1. (20 points) **Maximum-Subarray Problem:** Textbook Exercise 4.1-1 on p. 74.

   In problems 2-4, we solve the following recurrence in several different ways. Our goal is to find a tight upper and lower bound on the closed-form solution for this recurrence:

   \[
   T(n) = \begin{cases} 
   T\left(\frac{n}{3}\right) + T\left(\frac{n}{3}\right) + 2^{\log_3 n} & n > k \\
   \Theta(1) & n \leq k 
   \end{cases}
   \]

   where \( n = 3^j \) for a positive integer \( j \) and \( k \) is a small positive integer. That is, find a function \( g(n) \) such that \( T(n) \in \Theta(g(n)) \). (The \( \Theta(1) \) terminating condition is intended to represent some small constant.)

2. (20 points) Use the **Master Theorem** to solve this recurrence.

3. (20 points) Use a **recursion tree** to solve this recurrence.

4. (20 points) Use **substitution/induction** to prove the correctness of your answer to Problems (2, 3) above. Be sure to justify both the \( O \) and \( \Omega \) parts of the \( \Theta \) statement.

5. (20 points) **Pseudocode Analysis:** For the pseudocode below for Mystery\((n)\), derive tight upper and lower bounds on its asymptotic \textit{worst-case} running time \( T(n) \). That is, for the set of inputs including those that force Mystery to work its hardest, find \( g(n) \) such that \( T(n) \in \Theta(g(n)) \). Assume that the input \( n \) is a positive integer. Justify your answer.

   ```
   function Mystery(n):
   if n <= k:
       return \Theta(1)
   else:
       return T(n/3) + T(n/3) + 2^{\log_3 n}
   ```
Mystery (n)
1    if n \leq 1
2       return 1
3    else
4       for i = 1 to n
5          for j = i to 2n
6            print "Mystery print"
7    return Mystery (n/2)