CS 3510 - Honors Algorithms
Homework 1
Assigned January 18
Due Thursday, January 25

1. a) Give two functions f_1 and f_2 so that the following equation is true for f_1 and false for f_2:

$$\sum_{i=1}^{n} f(i) = \Theta(f(n)).$$

Prove your answers.

b) Prove that

$$\log(n!) = \Theta(n \log n)$$

from the basics (i.e., no not use Stirling's approximation of the factorial function, if you know this).

2. Solve the following recurrence relations. Give a Θ bound for each problem. Justify your answers (a few lines should suffice). You may assume in each case that $T(1) = O(1)$. Consider a change of variables if this helps.

a) $T(n) = T(\sqrt{n}) + 1$
b) $T(n) = 2T(n-1) + 1$
c) $T(n) = 2T(n/3) + 1$
d) $T(n) = 49T(n/25) + (\sqrt{n})^3 \log n$
e) $T(n) = 9T(n/3) + n^2 \log n$
f) $T(n) = 8T(n/2) + n^3$

3. a) Use a recursion tree (CLRS Exercise 4.2-4) to guess a solution and prove your answer by induction:

$$T(n) = T(n-d) + T(d) + cn$$

for constants $d \geq 1$ and $c > 0$. Assume that $T(n) = \Theta(1)$ for $n \leq d$.

b) Use a recursion tree (CLRS Excercise 4.2-5) to guess a solution, and prove your guess by induction:

$$T(n) = T(\alpha n) + T(1 - \alpha)n) + cn$$

for constants $0 < \alpha < 1$ and $c > 0$.

4. a) You are given $n > 1$ coins. One of the coins is lighter than the others, but otherwise indistinguishable. You have a scale and can put, in a weighing, *one* coin on each side of the scale. How many weightings do you need to find the fake coin? Give an upper-bound and a lower-bound argument.

   b) Same as part (a), only now you can but *any number* of coins on each side of the scale in a single weighing.

   c) Same as part (a), only now you can put up to k coins on each side of the scale in one weighing, for some constant $k \geq 1$.

5. **Extra credit: Optional!** You are given an array $A[1..n]$ of pairwise distinct integers.

   The *median* of A is the $n/2$-th largest integer (define the rounding as you like) in the array.

   A γ-approximate median for A is one of the integers that is at least as large as $\gamma n - 1$ of the numbers and no larger than $\gamma n - 1$ of the numbers in the array. (I.e., the true median is basically a $1/2$-approximate median.) You may assume $0 < \gamma \leq 1/2$.

   a) Given a linear time algorithms for finding a γ-approximate median, given a linear time algoirthm for finding the median. (I.e., use the linear time algorithm for the γ-approximate median as a subroutine. Assume each call takes linear time in the size of the array when you analyze the overall time of your algorithm.)

   b) Give a linear time algorithm for finding the median. (Hint: what is the median of the medians of $n/7$ disjoint groups of 7 elements each? Is it a γ-approximate median for some γ? If so, then you can use your solution to part (a) to solve part (b).)