CS314: Algorithms
Homework 0

Required Problems

1. Recurrences (20 points)

   Solve the following recurrences. State tight asymptotic bounds for each function in the form $\Theta(f(n))$ for some recognizable function $f(n)$. You do not need to turn in proofs (in fact, please *don't* turn in proofs), but you should do them anyway just for practice. Assume reasonable but nontrivial base cases if none are supplied. More exact solutions are better.

   (a) $A(n) = 2A(n/2) + \lg n$
   (b) $B(n) = 3B(n/2) + n$
   (c) $C(n) = 2C(n/2) + n^2$
   (d) $D(n) = 2D(n-1) + 1$
   (e) $E(n) = 2E(\lfloor n/3 \rfloor + 9) + n^2$
   (f) $F(n) = 2F(n-1)/F(n-2)$
   (g) $G(n) = G(n/2) + 1$

2. Sorting functions (20 points)

   Sort the following 25 functions from asymptotically smallest to asymptotically largest, indicating ties if there are any. You do not need to turn in proofs (in fact, please *don't* turn in proofs), but you should do them anyway just for practice.

1	n	n^2	$\lg n$	$1 + \lg\lg n$
$\cos n + 2$	$n^{\lg n}$	$(\lg n)!$	$(\lg n)^{\lg n}$	F_n
$\lg^{1000} n$	$2^{\lg n}$	$n \lg n$	$\sum_{i=1}^{n} i$	$\sum_{i=1}^{n} i^2$
$n!$	$\lg(n^{10000})$	$\lfloor \lg\lg(n) \rfloor$	$2^{2\log n}$	$15n^2 - 12n + 8\lg n + 4$

   To simplify notation, write $f(n) \ll g(n)$ to mean $f(n) = o(g(n))$ and $f(n) \equiv g(n)$ to mean $f(n) = \Theta(g(n))$. For example, the functions n^2, n, $\binom{n}{2}$, n^3 could be sorted either as $n \ll n^2 \equiv \binom{n}{2} \ll n^3$ or as $n \ll \binom{n}{2} \equiv n^2 \ll n^3$. *[Hint: When considering two functions $f(\cdot)$ and $g(\cdot)$ it is sometime useful to consider the functions $\ln f(\cdot)$ and $\ln g(\cdot)$.]*

3. Prove that any postage that is a positive number integer numbers of cents greater than 7 cents can be formed using just 3 cent stamps and 5 cent stamps. (Hint: Use induction, but remember your base cases!)

4. Describe an algorithm which implements a stack using two queues. Your algorithm should use ONLY two queues (and the appropriate operations on queues). What is the running time? (Note: Faster algorithms will be worth more credit!)