Algorithms CS315 **Practive Exam 1** **February 16, 2018**

1. (20 points) Answer each of the following short answer questions. Justify your answers in each case. These are to be short answers (truth) – not novels (fiction)!
   (a) (5 points) Let P be a problem. Suppose algorithm A is a solution to problem P with asymptotic worse case run time of $O(n^2)$. What can we say about the asymptotic worse case run time of other algorithms that solve problem P? What does the run time of algorithm A say about problem P? What doesn't A say about P.
   (b) (5 points) Suppose algorithm A solves problem P and has actual run time $20 \log_2 n$ for all inputs of size n. Suppose algorithm B is also a solution to problem P with actual run time of $20 \log_3 n$ for all inputs of size n. Is the asymptotic run time behavior of the two algorithms the same? What can we conclude when comparing algorithm A and B?
   (c) (5 points) Suppose algorithm A runs in $\Theta(n^3)$ time. What can we say about the input to A relative to its run time?
   (d) (5 points) Show $c \log_b n = \Theta(\log_2 n)$ for $c, b > 1$.

2. (10 points) Solve the following recurrence equations for which the Master Method applies. Show your work.
   (a) (5 points) $T(n) = 7T(n/2) + n^2$
   (b) (5 points) $T(n) = 2T(n/2) + n \log_2 n$

3. (10 points) The recurrence $T(n) = 7T(n/2) + n^2$ describes the running time of an algorithm A. A competing algorithm A' has running time of $T'(n) = aT'(n/4) + n^2$. What is the largest integer value of a such that A' is asymptotically faster than A. (In this case, asymptotically faster means a smaller polynomial degree than $T(n)$.) Justify your work.

4. (20 points) For each function state its best and worst case asymptotic run time with respect to n. Assume all arithmetic operates in constant time and that a single integer prints in constant time. Justify your answers for full credit. Each part is worth 5 points.
   (a) (5 points)

```
Function (A)
    n = A.length
    for j = 2 to n
        k = A[j]
        i = j-1
        while i>0 and A[i]>k
           A[i+1] = A[i]
           i = i-1
    A[i+1] = k
```

(b) (5 points)

```
Function(A)
    n = A.length
    for i = 1 to n
        Print(A[1..i])
```

(c) (5 points)

```
Function(A)
    n = A.length
    for i = 1 to n
        Print(A[i])
```

(d) (5 points)

```
Function(n)
    if n <= 1 then
         return 10
    else
       m = Function(n/4)
       h = Function(n/4)
       return f(n,m.h)   // where f(n,m.h) runs in linear time
```

Master Theorem Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n),$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows,
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log_2 n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.