Algorithms and Data Structures
Spring 2018
Exercises for Unit 5

1. The following exercises are to remind you about asymptotic notation and some of its intricacies.

   (a) Order the following functions by their asymptotic growth rate, and prove that this ordering is correct:

   $$n/1000 \quad n^{4/3} \quad 5^n \quad n \log n \quad n^{1-\varepsilon} \quad n/\log\log n \quad n^{4/3}\log n \quad (4/3)^n \quad n \cdot 2^{\sqrt{\log n}}$$

   Here, $0 < \varepsilon < 1$ is a fixed constant.

   (b) We noted in class that $f \in O(g)$ can be viewed as analogous to $f \le g$, and $f \in o(g)$ can be viewed as analogous to $f < g$, and similarly for $f \in \Omega(g)$ analogous to $f \ge g$ and $f \in \omega(g)$ analogous to $f > g$.

   - For numbers $x \not< y$ is the same as $x \ge y$. Is $f \notin o(g)$ the same as $f \in \Omega(g)$?
   - Is it possible that $f \notin o(g)$ and at the same time $g \notin o(f)$?
   - How about $f \notin O(g)$ and at the same time $g \notin O(f)$?

2. Use the Akkra-Bazi Theorem to derive closed form expressions for the following recursively defined functions:

   (a) $T(n) = 2 \cdot T(n/4) + \sqrt{n}$

   (b) $T(n) = 3 \cdot T(n/2) + n \log n$

   (c) $T(n) = T(n/2) + \alpha$

   (d) $T(n) = 4 \cdot T(n/2) + n^2/\log n$

   It is assumed that in all cases we have $T(n) = c \cdot n$ for $n \le 4$, and α and c are some positive constants.

3. Use the Akkra-Bazi Theorem to derive the "Master Theorem" for divide-and-conquer recurrence relations, which we stated in class and which you find in most textbooks on analysis of algorithms, stated in some form, e.g. as

   Theorem (Master Theorem)

   Let $T(n)$ be a monotonically increasing function that satisfies

   $$\begin{aligned} T(n) &= aT(\tfrac{n}{b}) + f(n) \\ T(1) &= c \end{aligned}$$

   where $a \ge 1, b \ge 2, c > 0$. If $f(n) \in \Theta(n^d)$ where $d \ge 0$, then

   $$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$