Little1b.Solution

PROBLEM 1 *Iterated Functions*

When solving recurrence relations, we typically need to determine the depth of the recursion. For instance, consider the Merge Sort Algorithm from class. The depth of our recursion in this case was $\log_2 n$ for a list of length n (i.e., the number of times we needed to divide n by 2 until hitting a base case—a value ≤ 1). For this problem, we will generalize this idea of iterative re-application of a function until a target value is reached. This will help us determine the recursive depth of algorithms.

Definition 1 *Define the iterated function $f_c^*(n) = \min\{i \geq 0 \mid f^{(i)}(n) \leq c\}$ where f is a monotonically increasing function over the reals, c is a fixed constant, and for an integer $i > 1$, $f^{(i)}(n) = f(f^{(i-1)}(n))$, e.g., $f^{(3)}(n) = f(f(f(n)))$.*

In the Merge Sort algorithm, we could describe the depth of our recursion using this new iterated function notation with $f(n) = \frac{n}{2}$ and $c = 1$, giving $f_1^*(n) = \lceil \log_2 n \rceil$.

Fill in the table below. For each of the given functions $f(n)$ and constants c, give as tight a bound as possible on $f_c^*(n)$. The first row is done for you. Justify each answer.

$f(n)$	c	$f_c^*(n)$
$n/2$	1	$\lceil \log_2 n \rceil$
$n-1$	0	$\lceil n \rceil$
$n/2$	2	$\lceil \log_2 n \rceil - 1$
$\sqrt{n}$	2	$\lceil \log_2 \log_2 n \rceil$
$\sqrt{n}$	1	∞
$n^{1/3}$	2	$\lceil \log_3 \log_2 n \rceil$

Justification:

2. If n is in the range $(0, 1]$ then $f_0^*(n) = 1 = \lceil n \rceil$. For any choice of n where $f_0^*(n) = x$, we have that $f_0^*(n+1) = x+1$ since $f_0^*(n+1) = f_0^*(f(n+1)) + 1 = f_0^*(n) + 1 = x + 1$
3. $f^{(k)}(n) = n/2^k$, so if we require $n/2^k \leq 2$, then $2^k \geq n/2$, and correspondingly, $k \geq \log_2(n/2) = \log_2(n) - 1$. Thus, $f_2^{(*)}(n) = \lceil \log_2 n - 1 \rceil = \lceil \log_2 n \rceil - 1$.
4. $f^{(k)}(n) = n^{1/2^k}$, so if we require $n^{1/2^k} \leq 2$, we have that $1/2^k \log_2 n \leq 1$. This means $2^k \geq \log_2 n$, so $k \geq \log_2 \log_2 n$ and $f_2^{(*)}(n) = \lceil \log_2 \log_2 n \rceil$.
5. $f^{(k)}(n) = n^{1/2^k}$, so we require $n^{1/2^k} \leq 1$. For all values of $k > 0$, we have that $1/2^k > 0$, so $n^{1/2^k} > 1$. As $k \to \infty$, then $1/2^k \to 0$, and $n^{1/2^k} \to 1$. Thus, $f_1^{(*)}(n) = \infty$.

6. $f^{(k)}(n) = n^{1/3^k}$, so if we require $n^{1/3^k} \leq 2$, we have that $1/3^k \log_2 n \leq 1$. This means $3^k \geq \log_2 n$, so $k \geq \log_3 \log_2 n$ and $f_2^{(*)}(n) = \lceil \log_3 \log_2 n \rceil$.

PROBLEM 2 *Asymptotics*

Given that:
$\forall \varepsilon > 0,\ \log(n) \in o(n^\varepsilon)$,
show:
$\forall \varepsilon, k > 0,\ \log^k(n) \in o(n^\varepsilon)$

Justification: Let $\varepsilon, k > 0$ be arbitrary constants. We need to show that for every constant $c > 0$, there exists $n_0 > 0$ such that for all $n > n_0$, $\log^k(n) < c \cdot n^\varepsilon$. Let $c' = c^{1/k}$ and $\varepsilon' = \varepsilon/k$. By assumption, there exists $n_0' > 0$ such that for all $n > n_0'$, $\log(n) < c' \cdot n^{\varepsilon'}$. Correspondingly, for $n > n_0'$, we have that

$$\log^k(n) < (c' n^{\varepsilon'})^k = (c^{1/k} n^{\varepsilon/k})^k = cn^\varepsilon.$$

Taking $c = c'$ and $n_0 = n_0'$ proves the claim.