Asymptotic Expressions

Here is a video walkthrough of the solutions.

(a) Which of the following expressions are true? Check all that apply. Equations between asymptotic expressions, such as $O(f) = O(g)$ simply mean that all functions that are $O(f)$ are also $O(g)$ and vice-versa. An expression such as $O(f) \subseteq O(g)$ means that all functions that are $O(f)$ are also $O(g)$.

- ☐ $\Theta(1000 * N^3 + N * \log(N)) = \Theta(N^3)$.
- ☐ For all $k \geq 0$, $O(N^k) \subseteq O(N^{k+1})$.
- ☐ For all $k \geq 0$, $\Omega(N^k) \subseteq \Omega(N^{k+1})$.
- ☐ For positive-valued functions f and g, if $f = \Omega(g)$ and $g = O(h)$, $f = \Omega(h)$.
- ☐ For positive-valued functions f and g, if $f = \Omega(g)$ and $h = O(g)$, $f = \Omega(h)$.

Solution:

- ■ $\Theta(1000 * N^3 + N * \log(N)) = \Theta(N^3)$.
  True, we ignore lower order terms.
- ■ For all $k \geq 0$, $O(N^k) \subseteq O(N^{k+1})$.
  True, every function that is $O(N^k)$ is also $O(N^{k+1})$ since $O(N^{k+1})$ is a less tight bound.
- ☐ For all $k \geq 0$, $\Omega(N^k) \subseteq \Omega(N^{k+1})$.
  False, a function that runs in $\Theta(N^k)$ runs in $\Omega(N^k)$ but not $\Omega(N^{k+1})$.
- ☐ For positive-valued functions f and g, if $f = \Omega(g)$ and $g = O(h)$, $f = \Omega(h)$.
  False, f and h are lower bounded by g, but we can't say anything their relation.
- ■ For positive-valued functions f and g, if $f = \Omega(g)$ and $h = O(g)$, $f = \Omega(h)$.
  True, f is lower bounded by g and g upper bounds h, so f is also lower bounded by h.

(b) For positive-valued functions $f_0 \dots f_k$, where we define $f_i(n) = 1 + f_{n\%i}(n)$ for $i \geq 1$ and $f_0(n) = 1$, which of the following are true? Check all that apply. Assume that $n > k$.

- ☐ The evaluation of $f_k(n)$ may run forever.
- ☐ $f_k(n) = \Omega(log(k))$, with respect to k.
- ☐ $f_k(n) = O(k)$, with respect to k.
- ☐ $f_k(n) = \Theta(1)$, with respect to n.
- ☐ If $n = k! - 1$, $f_k(n) = \Theta(k)$, with respect to k.

Solution:

- ☐ The evaluation of $f_k(n)$ may run forever.
  False, notice that $n\%i$ is bounded between 0 and $i-1$, so $f_k(n)$ will recurse on some function $f_i(n)$ where $i < k$, and eventually the base case must be hit.

- ☐ $f_k(n) = \Omega(log(k))$, with respect to k.
  False, $f_k(n)$ could take constant time, e.g. when $n = 2 \times k$.

- ■ $f_k(n) = O(k)$, with respect to k.
  True, see the last part for the worst case behavior of $f_k(n)$

- ■ $f_k(n) = \Theta(1)$, with respect to n.
  True, since $f_k(n)$ recurses on $f_{n\%k}(n)$, the remainder operation bounds $n\%k$ between 0 and $k-1$, which is independent of n.

- ■ If $n = k! - 1$, $f_k(n) = \Theta(k)$, with respect to k.
  True, notice that $k!$ is divisible by every number between 1 and k, so when $k! - 1$ is divided by any i between 1 and k, it will have remainder $i - 1$. As such, $f_k(n)$ will recurse on $f_{k-1}(n)$, which will recurse on $f_{k-2}(n)$, and so on until $f_0(n)$ is hit, taking linear time with respect to k.