HW1 Solution

1. Asymptotic Notations:

   (a) Prove that $\sum_{i=1}^{k} i^k = \Theta(n^{k+1})$.

   We need to show that n^{k+1} provides both an upper bound and a lower bound.
   For the upper bound, observe that:

   $$\begin{aligned}\sum_{i=1}^{n} i^k &= 1^k + 2^k + \cdots + n^k \\ &\le n^k + n^k + \cdots + n^k = n \cdot n^k = n^{k+1}.\end{aligned}$$

   This holds for all $n \geq 1$. Thus, $\sum_{i=1}^{n} i^k = O(n^{k+1})$.
   For the lower bound, we follow a similar strategy and drop half of the terms.

   $$\begin{aligned}\sum_{i=1}^{n} i^k &= 1^k + 2^k + \cdots + (\tfrac{n}{2})^k + \cdots + n^k \\ &\ge (n/2)^k + (n/2)^k + \cdots + (n/2)^k = n/2 \cdot (n/2)^k = (n/2)^{k+1}.\end{aligned}$$

   This holds for all $n \geq 1$. Thus, $\sum_{i=1}^{n} i^k = \Omega(n^{k+1})$.

   (b) Transitivity of little-o, i.e. if $f(n) = o(g(n))$ and $g(n) = o(h(n))$, then $f(n) = o(h(n))$.

   Consider any positive constant $c \in \mathbb{R}^+$. Let's define $c_0 = \sqrt{c}$. Since $g(n) = o(h(n))$, we know we can find a constant n_0 such that

   $$g(n) < c_0 h(n), \forall n > n_0.$$

   Likewise, since $f(n) = o(g(n))$, we can find another constant n_1 such that

   $$f(n) < c_0 g(n), \forall n > n_1.$$

   From this, we can conclude that:

   $$c_0 g(n) < c_0^2 h(n) = ch(n), \forall n > \max(n_0, n_1).$$

   But, since $f(n) = o(g(n))$, we also must have that:

   $$f(n) < c_0 g(n) < ch(n), \forall n > \max(n_0, n_1).$$

   This holds for any positive c. Therefore $f(n) = o(h(n))$.
   Alternatively, this may also be shown via an argument based on limits. Since $f(n) = o(g(n))$, we have $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Similarly, since $g(n) = o(h(n))$, we have $\lim_{n\to\infty} \frac{g(n)}{h(n)} = 0$. Multiplying the two and using the product law of limits, we get $\lim_{n\to\infty 0} \frac{f(n)}{g(n)} \frac{g(n)}{h(n)} = 0$, or equivalently, $\lim_{n\to\infty} \frac{f(n)}{h(n)} = 0$. Thus, $f(n) = o(h(n))$.

(c) Let $p(n) = a_0 + a_1 n + a_2 n^2 + \cdots + a_k n^k$ where $a_k > 0$. Show that $p(n) = \Theta(n^k)$.

We know that $p(n) = \Theta(n^k)$ if and only if $p(n) = O(n^k)$ and $p(n) = \Omega(n^k)$. Thus, we need to show that $p(n)$ is both $O(n^k)$ and $\Omega(n^k)$.

To show that $p(n) = O(n^k)$, we need to find a positive constant c such that for sufficiently large n, $p(n) \leq c\,n^k$. Rewriting the polynomial as $p(n) = a_k n^k + \sum_{i=0}^{k-1} a_i n^i$ and dividing by n^k, the inequality becomes:

$$a_k + \sum_{i=0}^{k-1} a_i n^{i-k} \leq c.$$

For sufficiently large n, this becomes:

$$\begin{aligned}
\lim_{n\to\infty} \big(a_k + \sum_{i=0}^{k-1} a_i n^{i-k}\big) &\leq c, \\
a_k + \lim_{n\to\infty} \big(\sum_{i=0}^{k-1} a_i n^{i-k}\big) &\leq c, \\
a_k + \big(\sum_{i=0}^{k-1} \lim_{n\to\infty} a_i n^{i-k}\big) &\leq c, \\
a_k \leq c \quad \Big(\text{since } \lim_{n\to\infty} a_i n^{i-k} &= 0 \text{ for } 0 \leq i \leq k-1\Big).
\end{aligned}$$

Thus, choosing $c \geq a_k$ will ensure that $p(n) \leq c\,n^k$ for sufficiently large n.

To show that $p(n) = \Omega(n^k)$, we need to find a positive constant c such that for sufficiently large n, $p(n) \geq c\,n^k$. Following the same procedure as above, we see that this will be the case if $c \leq a_k$.

Thus, since $p(n)$ is both $O(n^k)$ and $\Omega(n^k)$, $p(n) = \Theta(n^k)$.

(d) Find two positive valued functions $f(n)$ and $g(n)$ such that neither $f(n) = O(g(n))$ nor $g(n) = O(f(n))$.

We can find two positive-valued functions that oscillate relative to each other asymptotically. Once such example is:

$$f(n) = 1 + \cos(\pi n)$$
$$g(n) = 1 + \sin(\pi n + \tfrac{3\pi}{2})$$

2. Using a loop invariant or otherwise, determine the asymptotic complexity of the following loop. State your answer using an appropriate asymptotic notation. Please show all the steps that lead you to the asymptotic characterization.

```
sum = 0;
```

```
for( i = 1; i < n; i++ )
  for( j = 1; j < i * i; j++ )
    if( j % i == 0 )
      for( k = 0; k < j; k++ )
        sum++;
```

Let's try and come up with a loop-invariant for the outer-most loop by inspecting what happens for a couple of different values of `i`.

`i`	`j`	`j mod i == 0`	`sum`
2	$1, 2, 2^2 - 1$	2	`+=` 2
3	$1, 2, \ldots, 3^2 - 1$	$3, 6$	`+=` $3 + 6$
4	$1, 2, \ldots, 4^2 - 1$	$4, 8, 12$	`+=` $4 + 8 + 12$
$\vdots$			
i	$1, 2, \ldots, i^2 - 1$	$i, 2i, \ldots, (i-1)i$	`+=` $i + 2i + \cdots + (i-1)i$

Thus, after iteration i, the value of `sum` is incremented by $\sum_{m=1}^{i-1} m \cdot i$. The value of `sum` after the loop will therefore be:

$$\begin{aligned}
&\sum_{i=1}^{n-1} \sum_{m=1}^{i-1} m \cdot i \\
=&\sum_{i=1}^{n-1} \frac{i^2(i-1)}{2} \\
=&\Theta(n^4) \quad \text{(based on the answer to problem 1a).}
\end{aligned}$$

The summation above can be evaluated in closed form but this is not necessary if we just want an asymptotic characterization.

3. Weiss (Exercise 2.14) and CLRS (Problem 2-3): Horner's rule is an efficient method to evaluate a univariate polynomial $f(x) = \sum_{k=0}^{n} a_k x^k$. It is given by the following algorithm:

   $y \leftarrow 0$;
   for $i \leftarrow n$ **to** 0 **do**
   │ $y \leftarrow a_i + x \cdot y$;
   end

   (a) Trace the execution of this algorithm for $x = 3$ and $f(x) = 4x^4 + 8x^3 + x + 2$.

   For $f(x) = 2 + x + 8x^3 + 4x^4$, $n = 4$ and the coefficient sequence is $\boldsymbol{a} = (2, 1, 0, 8, 4)$. The different iterations are tabulated below.

i	y
4	4
3	$8 + 3 \cdot 4$
2	$0 + 3 \cdot (8 + 3 \cdot 4)$
1	$1 + 3 \cdot (0 + 3 \cdot (8 + 3 \cdot 4))$
0	$2 + 3 \cdot (1 + 3 \cdot (0 + 3 \cdot (8 + 3 \cdot 4)))$

(b) Consider the following loop invariant:
At the start of each iteration of the for loop:

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k.$$

Show that this loop invariant holds by showing the *initialization* and *maintenance* steps.

Initialization: Just before the loop starts, $i = n$. Substituting into the loop invariant, we get $y = \sum_{k=0}^{-1} a_{k+n+1} x^k = 0$ (empty sum). Since y is initialized to 0, the initialization step holds.
Maintenance: Suppose the loop invariant holds at the start of iteration i and the loop executes one more time. We need to show that the value of y at the start of the next iteration $(i-1)$ is $y = \sum_{k=0}^{n-i} a_{k+i} x^k$. In the loop, the value of y is updated to:

$$\begin{aligned} y &= a_i + x \cdot \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k \\ &= a_i x^0 + \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^{k+1} \\ &= a_i x^0 + \sum_{k=1}^{n-i} a_{k+i} x^k \\ &= \sum_{k=0}^{n-i} a_{k+i} x^k. \end{aligned}$$

The loop-invariant is therefore maintained.

(c) Use the loop invariant above to prove that the algorithm is correct.

We'll prove the partial correctness for the following specifications.
Pre-condition: $n \geq 0$ and the coefficient sequence $(a_0, \ldots, a_n)$ is given.
Post-condition: $y = \sum_{k=0}^{n} a_k x^k$.

Proof. From the loop invariant we showed above, we know that at the start of iteration i, $y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k$. The algorithm terminates after executing the $i = 0$ iteration. The next iteration would correspond to $i = -1$. Substituting $i = -1$ in the loop invariant, we obtain:

$$y = \sum_{k=0}^{n-(-1+1)} a_{k-1+1} x^k = \sum_{k=0}^{n} a_k x^k.$$

For complete correctness, we need to show that the algorithm terminates. This is straightforward since the for loop decrements i from n down to 0. □

(d) Analyze the running time of Horner's rule, state your answer in terms of Θ-notation.

> The for loop executes $n+1$ times. During each iteration (except for the first), there is one addition operation and one multiplication operation. Thus, the complexity of the algorithm is $\Theta(n)$ arithmetic operations. This is better than a straightforward evaluation of each of the monomials which requires $\Theta(n^2)$ multiplications.