Homework 1 Solutions
CSCI-2300: Data Structures and Algorithms
Sections 8, 9, 10

1. Weiss 1.8(a) and 1.8(b).

(a) \( \sum_{i=0}^{\infty} \frac{1}{4^i} \)

Solution:
\[
\sum_{i=0}^{\infty} \left( \frac{1}{4} \right)^i = \frac{1}{1 - \frac{1}{4}} = \frac{4}{3}
\]

(b) \( \sum_{i=0}^{\infty} \frac{i}{4^i} \)

Solution:
Let \( S = \sum_{i=0}^{\infty} \frac{i}{4^i} \)

Then \( S = \frac{1}{4^1} + \frac{2}{4^2} + \frac{3}{4^3} + \frac{4}{4^4} + \ldots \)

So,
\[
4S = 1 + \frac{2}{4^1} + \frac{3}{4^2} + \frac{4}{4^3} + \ldots
\]

Hence \( 4S - S = \left(1 + \frac{2}{4^1} + \frac{3}{4^2} + \frac{4}{4^3} + \ldots\right) - \left(\frac{1}{4^1} + \frac{2}{4^2} + \frac{3}{4^3} + \frac{4}{4^4} + \ldots\right) = \frac{4}{3} \), from (a).

Since \( 3S = \frac{4}{3} \), we get \( S = \frac{4}{9} \).

2. Evaluate the following summations, where \( a \) is a constant, \( a \neq 1 \).

(a) \( \sum_{i=10}^{n} (a^i + n) \)

Solution:
\[
\sum_{i=10}^{n} (a^i + n) = \sum_{i=0}^{n} (a^i + n) - \sum_{i=0}^{9} (a^i + n) = \left(\frac{a^{n+1} - 1}{a - 1} + (n + 1)n\right) - \left(\frac{a^{10} - 1}{a - 1} + 10n\right)
\]
\[
= \frac{a^{n+1} - a^{10}}{a - 1} + (n - 9)n
\]
(b) \[ \sum_{i=1}^{n-1} \left( a + \sum_{j=0}^{i-1} (j + 2) \right) \]

**Solution:**

A good solution strategy is to split the summation into subparts, and to evaluate nested summations by evaluating the innermost summations first.

Consider \( \sum_{j=0}^{i-1} (j + 2) \).

\[ \sum_{j=0}^{i-1} (j + 2) = \sum_{j=0}^{i-1} j + \sum_{j=0}^{i-1} 2 = \frac{(i-1)i}{2} + 2i = \frac{i^2 + 3i}{2}. \]

Therefore, \[ \sum_{i=1}^{n-1} \left( a + \sum_{j=0}^{i-1} (j + 2) \right) = \sum_{i=1}^{n-1} a + \sum_{i=1}^{n-1} \left( \frac{(i^2 + 3i)}{2} \right) \]

\[ = (n-1)a + \sum_{i=1}^{n-1} \left( \frac{i^2}{2} \right) + \sum_{i=1}^{n-1} \left( \frac{3i}{2} \right) = (n-1)a + \frac{(n-1)(n)(2n+4)}{12} + \frac{3(n-1)n}{4} \]

3. Prove using mathematical induction that for all \( n \geq 1 \)

(a) \[ \sum_{i=1}^{n} i(i!) = (n+1)! - 1 \]

**Solution:**

**Basis case:** For \( n = 1 \), the left hand side and the right hand side both evaluate to 1, proving the basis case.

**Induction step:** By the induction hypothesis, for all \( k \), \( 1 \leq k < n \), \( \sum_{i=1}^{k} i(i!) = (k+1)! - 1 \).

In particular, for \( k = n - 1 \), \( \sum_{i=1}^{n-1} i(i!) = (n)! - 1 \).

Therefore, \( \sum_{i=1}^{n} i(i!) = \sum_{i=1}^{n-1} i(i!) + n(n!) = (n)! - 1 + n(n!) = (n!)(n+1) - 1 = (n+1)! - 1 \), completing the proof by induction.

(b) \[ \sum_{i=1}^{n} i^3 = \left( \sum_{i=1}^{n} i \right)^2 \]

**Solution:**

**Basis case:** For \( n = 1 \), the left hand side and the right hand side both evaluate to 1, proving the basis case.

**Induction step:** By the induction hypothesis, for \( 1 \leq k < n \),

\[ \sum_{i=1}^{k} i^3 = \left( \sum_{i=1}^{k} i \right)^2 \]
Since, by the appropriate summation formula, \( \sum_{i=1}^{k} i = \frac{k(k + 1)}{2} \),

\[
\sum_{i=1}^{k} i^3 = \left( \frac{k(k + 1)}{2} \right)^2
\]

For \( k = n - 1 \),

\[
\sum_{i=1}^{n-1} i^3 = \left( \sum_{i=1}^{n-1} i \right)^2 = \left( \frac{(n - 1)n}{2} \right)^2
\]

Now \( \sum_{i=1}^{n} i^3 = \sum_{i=1}^{n-1} i^3 + n^3 = \left( \frac{(n - 1)n}{2} \right)^2 + n^3 = n^2 \left[ \frac{(n - 1)^2}{2^2} + n \right] \)

\[
= n^2 \left[ \frac{n^2 - 2n + 1 + 4n}{4} \right] = n^2 \left[ \frac{n^2 - 2n + 1 + 4n}{4} \right]
\]

\[
= n^2 \left[ \frac{n^2 + 2n + 1}{4} \right] = n^2 \left[ \frac{(n + 1)^2}{4} \right] = \frac{n^2(n + 1)^2}{2^2}
\]

Therefore, \( \sum_{i=1}^{n} i^3 = \left( \frac{n(n + 1)}{2} \right)^2 = \left( \sum_{i=1}^{n} i \right)^2 \)

4. For each pair of functions, \( T(n) \) and \( f(n) \), determine if \( T(n) = O(f(n)) \), \( T(n) = \theta(f(n)) \), both, or neither.

Justify your answers. (Assume \( a \) and \( b \) are unspecified constants greater than 1 and \( a > b \).)

**Important note:** \( T(n) = \theta(f(n)) \) always implies \( T(n) = O(f(n)) \), but \( T(n) = O(f(n)) \) does not by itself imply \( T(n) = \theta(f(n)) \).

(a) \( T(n) = n^2 \log n + 6n, f(n) = n^3 + n^2 \)

**Solution:**

First evaluate the limit of \( T(n)/f(n) \) as \( n \) tends towards \( \infty \) as shown below.

\[
\lim_{n \to \infty} \frac{T(n)}{f(n)} = \lim_{n \to \infty} \frac{n^2 \log n + 6n}{n^3 + n^2} = \lim_{n \to \infty} \frac{\log n + \frac{6}{n}}{1 + \frac{1}{n}} = 0
\]

Therefore, \( T(n) = O(f(n)) \).

Next evaluate the limit of \( f(n)/T(n) \) as shown below.

\[
\lim_{n \to \infty} \frac{f(n)}{T(n)} = \lim_{n \to \infty} \frac{n^3 + n^2}{n^2 \log n + 6n} \to \infty
\]
Therefore, \( f(n) \neq O(T(n)) \).
Since \( f(n) \neq O(T(n)) \), \( T(n) \neq \theta(f(n)) \).
Therefore, \( T(n) = O(f(n)) \), but \( T(n) \neq \theta(f(n)) \).
(b) \( T(n) = \log_a(n^4) \), \( f(n) = \log_b n \)

**Solution:**
Evaluate the limit of \( T(n)/f(n) \) as \( n \) tends towards \( \infty \) as shown below.

\[
T(n) = \log_a(n^4) = 4\log_a n \\

f(n) = \log_b n = \frac{\log_a n}{\log_a b}
\]

\[
\lim_{n \to \infty} \frac{T(n)}{f(n)} = \lim_{n \to \infty} \frac{4\log_a n}{\log_a b} = 4 \log_a b
\]

Since the limit evaluates to \( 4 \log_a b \), a non-zero constant, \( T(n) = \theta(f(n)) \). This implies that \( T(n) = O(f(n)) \). Therefore, \( T(n) = O(f(n)) \) and \( T(n) = \theta(f(n)) \).

5. Give the best possible “O” estimate for:

Justify your answers.

(a) \( T(n) \), where \( T(n) = 3^n \log n + (200n^2 + 15n^3) \cdot (n^3 + n \log n) \).

**Solution:**
\( T(n) = 3^n \log n + 200n^5 + 15n^6 + 200n^3 \log n + 15n^4 \log n \). The first term \( 3^n \log n \) is \( O(3^n \log n) \), the second term \( 200n^5 \) is \( O(n^5) \), the third term \( 15n^6 \) is \( O(n^6) \), the fourth term \( 200n^3 \log n \) is \( O(n^3 \log n) \), and the fifth term \( 15n^4 \log n \) is \( O(n^4 \log n) \). The dominant term is the first term, \( 3^n \log n \), since exponentials of base greater than 1 grow faster than polynomials. Hence \( T(n) = O(3^n \log n) \).

(b) the computation time required by the following code as a function of \( n \).

```c
sum=0;
for( i=0; i < n; i++)
    for( j=0; j < i*i; j++)
        for( k = 0; k < j; k++)
            sum++;
```

**Solution:**
\( O(n^5) \). The running time depends on the total number of iterations of the inner loop, where a constant time statement is executed. There are three nested for loops. The outermost for loop has \( n \) iterations. The middle for loop has \( i^2 \) iterations for each iteration of the outer loop. Therefore, the upper bound on the number of the iterations of the middle loop is \( n^2 \). The innermost for loop has \( j \) iterations for each iteration of the middle for loop. The upper bound on \( j \) is \( n^2 \) and hence bounds the number of iterations of the innermost for loop. Combining these upper bounds of the number of iterations of the for loops shows that the total number of iterations of the inner loop is bounded by \( n^5 \). Therefore, the running time of the code fragment is \( O(n^5) \).