Assignment 1

Quinn Perfetto, 104026025
60-454 Design and Analysis of Algorithms

January 27, 2017

Question 1 (i). We want to prove that when Algorithm Summation terminates its execution,

$$y = \sum_{j=0}^{n} a_j x^j$$

Proof. We shall apply induction to prove that after the mth iteration of the for loop the following invariant holds true:

$$y = \sum_{j=0}^{m-1} a_{n-j} x^{n-i-j}$$

(Induction basis) First, we note that y is initialized to 0 in Line 1. When $m = 1$ and $i = n$ in Line 3,

$$\begin{aligned} y &= a_n + x * y \\ &= a_n + x * 0 \\ &= a_n \\ &= a_n x^0 \\ &= \sum_{j=0}^{0} a_{n-j} x^{n-i-j} \end{aligned}$$

(Induction Hypothesis) Suppose for iteration $m - 1 < n$, $y = \sum_{j=0}^{m-2} a_{n-j} x^{n-i-j-1}$
(Induction Step) When Line 3 is executed for the mth time,

$$
\begin{aligned}
y &= a_{n-i} + x \sum_{j=0}^{m-2} a_{n-j} x^{n-i-j-1} \\
&= a_{n-i} + \sum_{j=0}^{m-2} a_{n-j} x^{n-i-j} \\
&= a_{n-i} x^0 + \sum_{j=0}^{m-2} a_{n-j} x^{n-i-j} && \text{(Induction Hypothesis)} \\
&= \sum_{j=0}^{m-1} a_{n-j} x^{n-i-j}
\end{aligned}
$$

Therefore we can conclude that the invariant $y = \sum_{j=0}^{m-1} a_{n-j} x^{n-i-j}$ holds $\forall m > 0$.

On the n+1th iteration (where $i = 0$) we then have,

$$
\begin{aligned}
y &= \sum_{j=0}^{n} a_{n-j} x^{n-j} \\
&= a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 x^0 \\
&= \sum_{j=0}^{n} a_j x^j && \text{(By reversing the order of the terms)}
\end{aligned}
$$

$\square$

Question 1 (ii).
Key Operation: Multiplication of real numbers.
Input size: n (The size of the input array)

The Summation algorithm must perform the key operation on each of the n elements of the input array.

1. Worst-case Time Complexity: $T(n) = n$
2. Average-case Time Complexity: $T_{ave}(n) = n$

Question 3. I assert that the claim is false, and will provide a counter example such that $f \in O(g) \wedge h(f) \notin O(h(g))$.

Proof. Let

$$f(n) = n^2,$$
$$g(n) = n^3,$$
$$h(n) = \frac{1}{n}$$

Lemma I. $f \in O(g)$, using Theorem 0.2

$$\lim_{n\to\infty} \frac{n^2}{n^3}$$
$$= \lim_{n\to\infty} \frac{1}{n}$$
$$= 0(\geq 0) \qquad \text{(Theorem 0.2)}$$

Given the defintion of f and h we can see that $h(f(n)) = \dfrac{1}{n^2}$.
Given the defintion of g and h we can see that $h(g(n)) = \dfrac{1}{n^3}$.
All that remains is to prove that $\frac{1}{n^2} \notin O(\frac{1}{n^3})$. Using Theorem 0.2,

$$\lim_{n\to\infty} \frac{\frac{1}{n^2}}{\frac{1}{n^3}}$$
$$= \lim_{n\to\infty} \frac{n^3}{n^2}$$
$$= \lim_{n\to\infty} n$$
$$= \infty(\notin \mathbb{R}^+)$$

Hence combining Lemma I with the above proof we have created a counter example such that,

$$f \in O(g) \wedge h(f) \notin O(h(g))$$

□

Question 4. I assert that the following function ordering respects the relationship $g1 = o(g2)$, $g2 = o(g3)$, ..., $g7 = o(g8)$.

$$10^{100},\ weirdsum,\ lg(n),\ 10^{lglg(n)},\ 2^{\sqrt{2lg(n)}},\ 4^{lgn},\ n^{lgn},\ 2^n$$

A series of proofs follow to confirm this ordering,

Proof. $n^{lgn} = o(2^n)$

Given that $n^k = o(2^n), \forall k > 0$ was proven on Page 48 of Chapter 0, and $lgn > 0, \forall n > 1$ we have $n^{lgn} = o(2^n)$. □

Proof. $4^{lgn} = o(n^{lgn})$

It can been seen that $4^{lgn} = n^{lg(4)} = n^2$. Using Theorem 0.2,

$$\lim_{n\to\infty} \frac{n^2}{n^{lgn}}$$

$$= \lim_{n\to\infty} \frac{\frac{n^2}{n^2}}{\frac{n^{lgn}}{n^2}}$$

$$= \lim_{n\to\infty} \frac{1}{n^{lgn-2}}$$

$$=0$$

Therefore $4^{lgn} = o(n^{lgn})$. □

Proof. $2^{\sqrt{2lgn}} \in o(4^{lgn})$

It can be seen that $4^{lgn} = (2^2)^{lgn} = 2^{2lgn}$. By letting x represent $2lgn$ and applying Theorem 0.2 we have,

$$\lim_{n\to\infty} \frac{2^{\sqrt{x}}}{2^x}$$

$$= \lim_{n\to\infty} \frac{\frac{2^{\sqrt{x}}}{2^{\sqrt{x}}}}{\frac{2^x}{2^{\sqrt{x}}}}$$

$$= \lim_{n\to\infty} \frac{1}{2^{x-\sqrt{x}}}$$

$$= 0$$

Therefore $2^{\sqrt{2lgn}} = o(4^{lgn})$. □

Proof. $lg(n) = o(10^{lglg(n)})$

It can be seen that $10^{lglg(n)} = lg^{lg10}n$, where $lg10 > 1 = 1 + \epsilon$ for some $\epsilon > 0$. Using

Theorem 0.2 we have,

$$\lim_{n\to\infty} \frac{lg(n)}{lg^{1+\epsilon}n}$$

$$= \lim_{n\to\infty} \frac{\frac{lg(n)}{lg(n)}}{\frac{lg^{1+\epsilon}n}{lg(n)}}$$

$$= \lim_{n\to\infty} \frac{1}{lg^{\epsilon}n}$$

$$=0$$

Therefore $lg(n) = o(10^{lglg(n)})$. □

Question 5 (a). $T(n) = 9T(\frac{n}{3}) + n^2lg(n) + 2n$

Proof. Using the general formula for recurrences we note that,

$$a = 9, b = 3, f(n) = n^2lg(n) + 2n$$

Lemma I. $f(n) \in \theta(n^2lg(n))$, using Theorem 0.2

$$\lim_{n\to\infty} \frac{n^2lg(n) + 2n}{n^2lg(n)}$$

$$= \lim_{n\to\infty} \frac{\frac{n^2lg(n)+2n}{n^2}}{\frac{n^2lg(n)}{n^2}}$$

$$= \lim_{n\to\infty} \frac{lg(n) + \frac{2}{n}}{lg(n)}$$

$$= \lim_{n\to\infty} \frac{lg(n)}{lg(n)}$$

$$= 1(>0)$$

By Lemma I we have $f(n) \in \theta(n^2 lg(n)) = \theta(n^{log_b a} log^k n)$ where $k = 1 (\geq 0)$. Therefore using Case 2 of the general recurrence forumla we have,

$$T(n) \in \theta(n^2 lg^2 n)$$

□

Question 5 (b). $T(n) = 3T(\frac{n}{3}) + \sqrt{n}$

Proof. Using the general formula for recurrences we note that,

$$a = 3, b = 3, f(n) = \sqrt{n}$$

Lemma 1. $f(n) \in O(n^{log_3 2})$

$$\begin{aligned} &\frac{1}{2} \leq log_3 2 \\ &\sqrt{n} \leq n^{log_3 2} && (n \geq 0) \end{aligned}$$

Therefore for c = 1, and $n_0 = 0$ we have,

$$\sqrt{n} <= cn^{log_3 2}, \forall n_0 > 0$$

Hence $f(n) \in O(n^{log_3 2})$.

By Lemma I we have $f(n) \in O(n^{log_b a - \epsilon})$ where $\epsilon = 1 (> 0)$. Therefore using Case 1 of the general recurrence forumla we have,

$$T(n) \in \theta(n^{log_3 3}) = \theta(n)$$

□

Question 5 (c). $T(n) = 8T(\frac{n}{4}) + n^2 lg^2 n$

Proof. Using the general formula for recurrences we note that,

$$a = 8, b = 4, f(n) = n^2 lg^2 n$$

Lemma I. $f(n) \in \Omega(n^{log_4 9})$

$$\begin{aligned} log_4 9 &\leq 2 \\ n^{log_4 9} &\leq n^2 \\ &\leq n^2 lg^2 n && (n \geq 2) \end{aligned}$$

Therefore for $c = 1$ and $n_0 = 2$ we have,

$$n^2 lg^2 n \geq cn^{log_4 9}$$

Hence $f(n) \in \Omega(n^{log_4 9})$.
By Lemma I we have $f(n) \in \Omega(n^{log_b a + \epsilon})$ where $\epsilon = 1$. Moreover, for sufficiently large n,

$$af(\frac{n}{b}1$$

□