CPSC5210 Fall 2018 Homework 3

Jamion Williams

October 4, 2018

1 Using the Elimination Method:

Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

USING the elimination method, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = T(n-1) + n$$

...has a second "iteration" of:

$$T(n-1) = T(n-2) + n - 1$$

...in which the left hand side ($T(n-1)$) cancels out part of the initial right-hand side's equation. In this instance (without a coefficient), we can remember that, without loss of generality,

$$n = a^k$$

...where a is the coefficient of the subproblem, in this case, 1.

As we cancel out iteratively, we are left with

$$T(n) = (n-1) + (n-2) + (n-3) + ... + k$$
$$T(n) = (1-1) + (1-2) + (1-3) + ... + k$$
$$T(1) = 1$$

which is a basic summation resulting in a Recursive Time Complexity of

$$T(n) = \frac{n(n-1)}{2}$$

Remembering that our definition of Big-O is where c_0, n_0 such that $\exists f(n) \leq c_0 * g(n); (\forall n > n_0)$, if $f(n) = log_2(9n)$ and $g(n) = O(log_2 n)$, and considering that the Big-O of the above summation is common, we have

$$O(f(n)) = n^2$$

or, more appropriately notated,

$$f(n) \in O(n^2)$$

2 Solve the below...

2.1 Using the Master Theorem:

Give the tight asymptotic bounds for $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$.

REMEMBERING that the Master Theorem dictates that

$$T(n) = aT(\frac{n}{b}) + f(n)$$

...we can take the recursive problem

$$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$$

...and assign $a = 2$, $b = 4$, and $f(n) = \sqrt{n}$

And further remembering that the Master Theorem compares $n^{log_b\, a}$ against $f(n)$ as follows:

CASE 1) If $n^{log_b\, a}$ is polynomially larger than $f(n)$, then $T(n) = \theta(n^{log_b\, a})$

CASE 2) If $n^{log_b\, a}$ is equal to $f(n)$, then $T(n) = \theta(f(n)\; lg\; n)$

CASE 3) If $f(n)$ is polynomially larger than $n^{log_b\, a}$, then $T(n) = \theta(f(n))$

...we can see that:

$$n^{log_b\, a} \stackrel{?}{=} f(n)$$

$$n^{log_4\, 2} \stackrel{?}{=} \sqrt{n}$$

$$\therefore n^{0.5} = \sqrt{n}$$

THEREFORE the abovementioned CASE 2 applies, meaning that the asymptotic bounds of

$$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$$

are:

$$f(n) = \theta(\sqrt{n}\; lg\; n)$$

2.2 Using Elimination:

USING the elimination method, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$$

...has further iterations of

$$2^1(T(\frac{n}{4^1}) = 2^2T(\frac{n}{4^2}) + \sqrt{\frac{n}{4^1}})$$

$$2^2(T(\frac{n}{4^2}) = 2^3T(\frac{n}{4^3}) + \sqrt{\frac{n}{4^2}})$$

$$2^3(T(\frac{n}{4^3}) = 2^4T(\frac{n}{4^4}) + \sqrt{\frac{n}{4^3}})$$

...in which the left hand side cancels out the T form of each previous iteration's right-hand side's equation. As we cancel out iteratively, we are left with

$$(2^1\sqrt{\frac{n}{4^1}}) + (2^2\sqrt{\frac{n}{4^2}}) + (2^3\sqrt{\frac{n}{4^3}}) + (2^k\sqrt{\frac{n}{4^k}})$$

which reduces down to

$$\sqrt{n} + \sqrt{n} + \sqrt{n} + ... + k$$

Remembering that, without loss of generality, $n = 2^k$ and $k = lg\ n$, this gives us a loose upper bound of

$$f(n) \in O(\sqrt{n}\ lg\ n)$$

3 Using the Master Theorem and Elimination Method:

Give asymptotic upper and lower bounds for $T(n)$ for each of the following.

FOR THE BELOW, we apply the Master Theorem as defined below:

WHERE

$$T(n) = aT(\frac{n}{b}) + f(n)$$

CASE 1) If $n^{log_b\ a}$ is polynomially larger than $f(n)$, then $T(n) = \theta(n^{log_b\ a})$
CASE 2) If $n^{log_b\ a}$ is equal to $f(n)$, then $T(n) = \theta(f(n)\ lg\ n)$
CASE 3) If $f(n)$ is polynomially larger than $n^{log_b\ a}$, then $T(n) = \theta(f(n))$

3.1 a)

$$T(n) = 2T(\frac{n}{2}) + n^4 \; \forall n > 2$$

ACCORDING to the Master Theorem, $a = 2$, $b = 2$, and $f(n) = n^4$. Therefore

$$n^{log_b\, a} \stackrel{?}{=} f(n)$$
$$n^{log_2\, 2} \stackrel{?}{=} n^4$$
$$n^1 < n^4$$

THEREFORE our CASE 3 applies, meaning that

$$f(n) = \theta(n^4)$$

USING ELIMINATION, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = 2T(\frac{n}{2}) + n^4$$

...has further iterations of

$$2^1(T(\frac{n}{2^1}) = 2^2T(\frac{n}{2^1}) + n^{4+1})$$
$$2^2(T(\frac{n}{2^2}) = 2^3T(\frac{n}{2^2}) + n^{4+2})$$
$$2^3(T(\frac{n}{2^3}) = 2^4T(\frac{n}{2^3}) + n^{4+3})$$

...in which the left hand side cancels out the T form of each previous iteration's right-hand side's equation. As we cancel out iteratively, we are left with

$$2^k(n^{4+k})$$

Remembering that, without loss of generality, $n = 2^k$ and $k = lg\ n$, this reduces to a recursive time complexity of

$$T(n) = n(n^{4+lg\ n})$$
$$T(n) = n^5 + n^{lg\ n}$$

Remembering that our definition of Big-O is where c_0, n_0 such that $\exists f(n) \leq c_0 * g(n); (\forall n > n_0)$, if $f(n) = log_2(9n)$ and $g(n) = O(log_2 n)$, we have

$$f(n) \in O(n^5)$$

3.2 d)

$$T(n) = 7T(\frac{n}{3}) + n^2 \ \forall n > 2$$

ACCORDING to the Master Theorem, $a = 7$, $b = 3$, and $f(n) = n^2$. Therefore

$$n^{log_b\, a} \stackrel{?}{=} f(n)$$
$$n^{log_3\, 7} \stackrel{?}{=} n^2$$
$$n^{1.77} < n^2$$

THEREFORE our CASE 3 applies, meaning that

$$f(n) = \theta(n^2)$$

USING ELIMINATION, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = 7T(\frac{n}{3}) + n^2$$

...has further iterations of

$$7(T(\frac{n}{3}) = 7^2T(\frac{n}{3^2}) + n^{2+1})$$
$$7^2(T(\frac{n}{3^2}) = 7^3T(\frac{n}{3^3}) + n^{2+2})$$
$$7^3(T(\frac{n}{3^3}) = 7^4T(\frac{n}{3^4}) + n^{2+3})$$

...in which the left hand side cancels out the T form of each previous iteration's right-hand side's equation. As we cancel out iteratively, we are left with

$$7^k * n^{2+k}$$

Remembering that, without loss of generality, $n = 7^k$ and $k = log_7\, n$, this reduces to a recursive time complexity of

$$T(n) = n * n^2 + n^{log_7\, n}$$

Looking at the largest element, this gives us a loose upper bound of:

$$f(n) \in O(n^2)$$

3.3 f)

$$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$$

ACCORDING to the Master Theorem, $a = 2$, $b = 4$, and $f(n) = \sqrt{n}$. Therefore

$$n^{log_b\ a} \stackrel{?}{=} f(n)$$
$$n^{log_4\ 2} \stackrel{?}{=} \sqrt{n}$$
$$n^{0.5} = \sqrt{n}$$

THEREFORE our CASE 2 applies, meaning that

$$f(n) = \theta(\sqrt{n}\ lg\ n)$$

USING ELIMINATION, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$$

...has further iterations of

$$2^1(T(\frac{n}{4^1}) = 2^2T(\frac{n}{4^2}) + \sqrt{\frac{n}{4^1}})$$
$$2^2(T(\frac{n}{4^2}) = 2^3T(\frac{n}{4^3}) + \sqrt{\frac{n}{4^2}})$$
$$2^3(T(\frac{n}{4^3}) = 2^4T(\frac{n}{4^4}) + \sqrt{\frac{n}{4^3}})$$

...in which the left hand side cancels out the T form of each previous iteration's right-hand side's equation. As we cancel out iteratively, we are left with

$$(2^1\sqrt{\frac{n}{4^1}}) + (2^2\sqrt{\frac{n}{4^2}}) + (2^3\sqrt{\frac{n}{4^3}}) + (2^k\sqrt{\frac{n}{4^k}})$$

which reduces down to

$$\sqrt{n} + \sqrt{n} + \sqrt{n} + ... + k$$

Remembering that, without loss of generality, $n = 2^k$ and $k = lg\ n$, this gives us a loose upper bound of

$$f(n) \in O(\sqrt{n}\ lg\ n)$$

4 Using the Master Theorem and Elimination Method:

Give asymptotic upper and lower bounds for $T(n)$ when $T(n) = 4T(\frac{n}{3}) + n\ log\ n$.

REMEMBERING that the simplified definition of the Master Theorem is:

$$T(n) = aT(\frac{n}{b}) + f(n)$$

CASE 1) If $n^{log_b\ a}$ is polynomially larger than $f(n)$, then $T(n) = \theta(n^{log_b\ a})$
CASE 2) If $n^{log_b\ a}$ is equal to $f(n)$, then $T(n) = \theta(f(n)\ lg\ n)$
CASE 3) If $f(n)$ is polynomially larger than $n^{log_b\ a}$, then $T(n) = \theta(f(n))$
ACCORDING to the Master Theorem, $a = 4$, $b = 3$, and $f(n) = n\ log\ n$. Therefore

$$n^{log_b\ a} \stackrel{?}{=} f(n)$$

$$n^{log_3\ 4} \stackrel{?}{=} n\ log\ n$$

$$n^{1.24} > n\ log\ n$$

THEREFORE our CASE 1 applies, meaning that

$$f(n) = \theta(n^{log_b\ a})$$

USING ELIMINATION, we aim to drive towards $T(1)$ by making the left hand side of the equation cancel out a portion of the right hand side. Therefore, the starting equation

$$T(n) = 4T(\frac{n}{3}) + n\ log\ n$$

...has further iterations of

$$4^1T(\frac{n}{3^1}) = 4^2T(\frac{n}{3^1}) + \frac{4^1}{3^1}n\ log\ n$$

$$4^2T(\frac{n}{3^1}) = 4^3T(\frac{n}{3^2}) + \frac{4^2}{3^2}n\ log\ n$$

$$4^3T(\frac{n}{3^2}) = 4^4T(\frac{n}{3^3}) + \frac{4^3}{3^3}n\ log\ n$$

...in which the left hand side cancels out the T form of each previous iteration's right-hand side's equation. As we cancel out iteratively, we are left with

$$T(n) = \frac{4^k}{3^k}n\ log\ n$$

Remembering that, without loss of generality, $n = 3^k$ and $k = log_3\ n$, this gives us a loose upper bound of

$$T(n) = \frac{nk}{n} n \; log \; n$$

$$T(n) = k * n \; log \; n$$

$$T(n) = log_3 \; n * log \; n$$

or, looking at the largest element,

$$f(n) \in O(log \; n)$$