Homework TA1 **Zankhanaben Ashish Patel**
COMP6651 **40067635**
Prof. Tiberiu Popa Due Date: 22/02/19

1 Prove Or Disprove

a)

$O(2^n) = O(4^n)$: **False**
First, we'll check for $2^n = O(4^n)$
$\therefore 2^n \leq c_1 \cdot 4^n$
$\implies c_1 = 1$
$\therefore 2^n \in O(4^n)$
Now, we'll check for $4^n = O(2^n)$
$\therefore 4^n \leq c_2 \cdot 2^n$
$\therefore 2^n \leq c_2 \cdot 1$
Which is not possible because c is a constant, we cannot find exact value for c .
$\therefore 4^n \notin O(2^n)$

$$\therefore O(2^n) \neq O(4^n)$$

b)

$O(\log(2^n)) = O(\log(4^n))$: **True**
First, we'll check for $\log(2^n) = O(\log(4^n))$
$\therefore \log(2^n) \leq c_1 \cdot \log(4^n)$
$\therefore n \log 2 \leq 2n \cdot c_1 \cdot \log 2$
$\implies c_1 = 1$
$\therefore \log(2^n) \in O(\log(4^n))$
Now, we'll check for $\log(4^n) = O(\log(2^n))$
$\therefore \log(4^n) \leq c_2 \cdot \log(2^n)$
$\therefore \log(2^{2n}) \leq c_2 \cdot \log(2^n)$
$\therefore 2n \leq c_2 \cdot n$
$\implies c_2 = 3$
$\therefore \log(4^n) \in O(\log(2^n))$

$$\therefore O(\log(2^n)) = O(\log(4^n))$$

c)

$O(n!) = O((n+1)!)$:**False**
First, we'll check for $n! = O((n+1)!)$
$\therefore n! \leq c_1 \cdot (n+1)!$
Which will always true for $c_1 = 1$
$\therefore n! \in O((n+1)!)$
Now, we'll check for $(n+1)! = O(n!)$
$\therefore (n+1)! \leq c \cdot n!$
$\therefore (n+1) \cdot n! \leq c \cdot n!$
$\therefore (n+1) \leq c \cdot 1$
Which is not possible because c is a constant, we cannot find exact value for c.
In $(n+1)!$ there are $(n+1) * n!$ operations. So it's $n+1$ operation more than $n!$.

Means $(n+1)! \notin O(n!)$
$$\therefore O(n!) \neq O((n+1)!)$$

2 Solve the following problems from the textbook :

<u>Exe : 3.1-1</u>

According to the definition of θ notation,

$$0 \leq c_1 \cdot (f(n) + g(n)) \leq max(f(n), g(n)) \leq c_2 \cdot (f(n) + g(n)) \tag{1}$$

By solving all inequalities, we can prove our problem.
Let,
$$f(n) \leq max(f(n), g(n))$$
$$g(n) \leq max(f(n), g(n))$$

$$\therefore (f(n) + g(n))/2 \leq max(f(n), g(n)) \tag{2}$$

Thus,
$$c_1 \cdot (f(n), g(n)) \leq max(f(n), g(n)) \tag{3}$$

using equations (2) and (3), $c_1 = 1/2$.
It is always true that,

$$max(f(n), g(n)) \leq (f(n) + g(n)) \tag{4}$$

$$max(f(n), g(n)) \leq c_2 \cdot (f(n) + g(n)) \tag{5}$$

Using equations (4) and (5), $c_2 = 1$.
From above derivations, We get,

$$0 \leq 1/2 \cdot (f(n) + g(n)) \leq max(f(n), g(n)) \leq 1 \cdot (f(n) + g(n))$$

hence proved,
$$max(f(n), g(n)) = \Theta(f(n) + g(n))$$

<u>Exe : 3.1-2</u>

As given,
$$0 \leq c_1 \cdot n^b \leq (n+a)^b \leq c_2 \cdot n^b, \forall n \geq n_0.$$

For $c_1 \cdot n^b \leq (n+a)^b$
Let, $c_1 = 1$
$$\therefore n^b \leq (n+a)^b, \forall n \geq 1$$

Now, for $(n+a)^b \leq c_2 \cdot n^b$

$$n^b \cdot (1 + \tfrac{a}{n})^b \leq (1+a)^b \cdot n^b, \forall n \geq 1$$

$\therefore$ Real constants,
$$c_1 = 1, c_2 = (1+a)^b, \forall n \geq 1.$$

Exe : 3.2-4

According to Cormen book, function $f(n)$ is polynomically bounded if $f(n) = O(n^k)$ for some constant k.

For, $\lceil \lg n \rceil!$

By taking proof of contradiction, let's assume that our function is polynomically bounded, where exists constant $c, kandn_0$.

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

$$\therefore f(n) \leq c \cdot n^k$$

$$\lceil \log n \rceil! \leq c \cdot n^k$$

By taking $n = 2^b$,

$$\lceil \log 2^b \rceil! \leq c \cdot (2^{k \cdot b})$$

$$\therefore b! \leq c \cdot (2^{k \cdot b})$$

Above equation will never true.

Since, factorial function is not exponentially bounded,and we cannot find such c that makes our condition true.

$\therefore \lceil \lg n \rceil!$ **is not polynomically bounded**.

For, $\lceil \lg \lg n \rceil!$

Again taking proof of contradiction,

Assume $n = 2^{2^k}$,

$$\lceil \lg \lg 2^{2^k} \rceil! \leq c \cdot (2^{2^k})$$

$$\therefore k! \leq c \cdot 2^{2^k}$$

To prove prove above equation, let's take log both side,

$\therefore log(k!) \leq 2^k$, because $\log_2 2 = 1$.

Now let's consider below inequalities to prove above equation to be true.

$$\log(k!) < \log(k^k) < k \log k < 2^k$$

Which is true $\forall k \geq 1$, and $c = 1$.

$\therefore \lceil \lg \lg n \rceil!$ **is polynomically bounded.**

Exe : 3.4

a)

Answer : False

$$f(n) \leq c \cdot g(n) \tag{6}$$

$$g(n) = O(f(n)) \tag{7}$$

$\exists c$ such that (6) holds true.

Now if $g(n) = O(f(n)) \implies g(n) = c \cdot (f(n))$

To make this condition to be true, there is not exist any c to make both equation true together.

$\therefore$ Let's prove by example,

$2^n = O(3^n)$ but $3^n \neq O(2^n)$

b)

Answer : False
Let's prove by contradiction. Considering given conjecture to be true.

$$\therefore c_1 \cdot min(f(n) + g(n)) \leq (f(n) + g(n)) \leq c_2 \cdot min(f(n) + g(n))$$

For, $(f(n) + g(n)) \leq c_2 \cdot min(f(n) + g(n))$
Above conjecture cannot be true for $f(n) = n^3$ and $g(n) = n$.

$$n^3 + n \leq c_2 \cdot min(n^3, n) = c_2 \cdot n$$

Which will never turns into true.

c)

Answer : True
$f(n) = O(g(n), \exists c, n_0$ such that
$f(n) \leq c \cdot g(n)$ and $n \geq n_0$. $(f(n) \geq 1)$ is given.
Taking log both side,
$\log(f(n)) \leq \log(c \cdot g(n))$

$$\log(f(n)) \leq \log c + \log(g(n)) \tag{8}$$

As given $f(n) \geq 1$ and $\lg(g(n)) \geq 1$. Both preserve after taking log also.
So, we need to find k such that, Using (8) where,

$$k \log(g(n)) \geq \log c + \log(g(n))$$

Since $\log(g(n)) \geq 1$

$$\therefore k \geq \log c + 1$$

So, $f(n) \geq k \log(g(n))$ will always true.

d)

Answer : False Let's assume $f(n) = 2n$ and $g(n) = n$
For above scenario $f(n) = O(g(n))$ is true.
Now, Let's check for

$$2^{f(n)} = O(2^{g(n)})$$

$$2^{f(n)} \leq c \cdot 2^{g(n)}, \forall n \geq n_0$$

$$2^{2n} \leq c \cdot 2^n$$

$$4^n \leq c \cdot 2^n$$

$$2^n \leq c, \forall n \geq n_0$$

Which cannot be possible since c is constant.
$\therefore$ This conjecture is False.

e)

Answer : False.
Let,

$$f(n) = \frac{1}{n}$$
$$\therefore \frac{1}{n} \leq c \cdot \frac{1}{n^2}$$
$$1 \leq c \cdot \frac{1}{n}$$

We have to find $n \geq n_0$ and c such that these inequalities always hold. Since c is constant and n is variable, we cannot find threshold point, such that this condition always hold. So it's false.

f)

Answer : True
For,

$$f(n) = O(g(n))$$
$$f(n) \leq c \cdot g(n)$$
$$\therefore g(n) \geq \frac{f(n)}{c} \tag{9}$$

For,

$$g(n) = \Omega(f(n))$$
$$g(n) \geq c \cdot f(n) \tag{10}$$

By comparing both equations (9) and (10)
$g(n)$ is always bigger.So this conjecture is true.

g)

Answer: False
By taking proof of contradiction, let's assume that given conjecture is true.

$$\therefore 0 \leq c_1 \cdot f(\frac{n}{2}) \leq f(n) \leq c_2 \cdot f(\frac{n}{2}), \exists c_1, c_2 \, and \, n \geq n_0$$

Let's take $f(n) = 2^{2n}$. (Same as 3.4 d)

$$\therefore c_1 \cdot 2^{\frac{2n}{2}} \leq 2^{2n} \leq c_2 \cdot 2^{\frac{2n}{2}}$$
$$c_1 \cdot 2^{\frac{2n}{2}} \leq 4^n \leq c_2 \cdot 2^n$$

Now, $4^n \leq c_2 \cdot 2^n$ will never holds true for any $c_2 \, and \, n \geq n_0$. Therefore, given conjecture is false.

Exe : 4.5-1

a)

$T(n) = 2\,T(\frac{n}{4}) + 1$
$\implies a = 2, b = 4, f(n) = 1, n^{\log_b a} = n^{\frac{1}{2}} = \sqrt{n}$
Since,
$f(n) = O(n^{\log_b a - \epsilon})$
$\quad = O(n^{\log_4 2 - \frac{1}{2}})$ for $\epsilon = \frac{1}{2}$.
We can apply master theorem case-1.
To conclude the solution, $T(n) = \Theta(n^{\log_b a})$

$$T(n) = \Theta(\sqrt{n})$$

b)

$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$.
$n = 2, b = 4, f(n) = n^{\frac{1}{2}}, n^{\log_4 2} = n^{\frac{1}{2}}$
Case-2 of master theorem will apply since,
$f(n) = \Theta(n^{\log_b a}) = \Theta(n^{\frac{1}{2}})$
Thus, the solution to the recurrence is,

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(\sqrt{n} \lg n)$$

c)

$T(n) = 2T(\frac{n}{4}) + n$.
$n = 2, b = 4, f(n) = n\,, n^{\log_4 2} = n^{\frac{1}{2}} = O(\sqrt{n})$
Since $f(n) = \Omega(n^{\log_4 2 + \epsilon})$ Where $\epsilon = \frac{1}{2}$
Case-3 Applies, Sufficiently large n and $c < 1$ we'll check regularity condition.
$af(\frac{n}{b}) = 2(\frac{n}{4}) = \frac{n}{2} \leq c \cdot f(n)$
As $f(n) = n$, We can take $c = \frac{1}{2}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n)$$

d)

$T(n) = 2T(\frac{n}{4}) + n^2$.
$n = 2, b = 4, f(n) = n^2, n^{\log_4 2} = n^{\frac{1}{2}}$
Since $f(n) = \Omega(n^{\log_4 2 + \epsilon})$ Where $\epsilon = \frac{3}{2}$
Case-3 Applies, Sufficiently large n and $c < 1$ we'll check regularity condition.
$af(\frac{n}{b}) = 2(\frac{n}{4})^2 = \frac{n}{8} \leq c \cdot f(n)$
As $f(n) = n^2$, We can take $c = \frac{1}{8}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n^2)$$

Exe : 4.5-5

Let's take $a = 1, b = 5, f(n) = 5n$ So, $n^{\log_5 1} = 0$.
Since $f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon = 1$

Case-3 Applies, Sufficiently large n and $c < 1$ we'll check regularity condition.
$af(\frac{n}{b}) = 1(\frac{5n}{5}) = n \leq c \cdot f(n)$
As $f(n) = 5n$, We will have $c = 1$ which is **not fulfilling** regularity condition which is $c < 1$.

Problem : 4.1

a)

$T(n) = 2T(\frac{n}{2}) + n^4$
Using masters theorem,
$a = 2, b = 2, f(n) = n^4, n^{\log_b a} = n^{\log_2 2} = n$
Since, $f(n) = \Omega(n^{\log_2 2+\epsilon}), \epsilon = 3$
Case-3 applies, Let's prove regularity condition for $f(n)$, for sufficient large value of n and $c < 1$.
$af(\frac{n}{b}) = 2(\frac{n}{2})^4 = \frac{n^4}{8} \leq c \cdot f(n)$
As $f(n) = n^4$, We can take $c = \frac{1}{8}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n^4)$$

b)

$T(n) = T(\frac{7n}{10}) + n$
Using masters theorem,
$a = 1, b = \frac{10}{7}, f(n) = n, n^{\log_b a} = n^{\log_{\frac{10}{7}} 1} = 0$
Since, $f(n) = \Omega(n^{\log_{\frac{10}{7}} 1+\epsilon}), \epsilon = 1$
Case-3 applies, Let's prove regularity condition for $f(n)$, for sufficient large value of n and $c < 1$.
$af(\frac{n}{b}) = 1 \cdot (\frac{7n}{10}) = \frac{7}{10}n \leq c \cdot f(n)$
As $f(n) = n$, We can take $c = \frac{7}{10}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n)$$

c)

$T(n) = 16T(\frac{n}{4}) + n^2$
Using masters theorem,
$a = 16, b = 4, f(n) = n^2, n^{\log_b a} = n^{\log_4 16} = n^2$
Case-2 applies,Since,
$f(n) = \Theta(n^{\log_4 16}) = \Theta(n^2)$ Thus, By case-2 the solution to the recurrence is,

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(n^2 \lg n)$$

d)

$T(n) = 7T(\frac{n}{3}) + n^2$
By master theorem,
$a = 7, b = 3, f(n) = n^2, n^{\log_b a} = n^{\log_3 7} \approx n^{1.77}$
Since, $f(n) = \Omega(n^{\log_3 7+\epsilon}), \epsilon \approx 0.23$
Case-3 applies, Let's prove regularity condition for $f(n)$, for sufficient large value of n and $c < 1$.
$af(\frac{n}{b}) = 7 \cdot (\frac{n}{3})^2 = \frac{7}{9}n^2 \leq c \cdot f(n)$

As $f(n) = n^2$, We can take $c = \frac{7}{9}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n^2)$$

e)

$T(n) = 7T(\frac{n}{2}) + n^2$
By master theorem,
$a = 7, b = 2, f(n) = n^2, n^{\log_b a} = n^{\log_2 7} \approx n^{2.8}$
Since, $f(n) = O(n^{\log_2 7 - \epsilon}), \epsilon \approx 0.8$
We can apply case-1,

$$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\lg 7})$$

f)

$T(n) = 2T(\frac{n}{4}) + \sqrt{n}$
By master theorem,
$a = 2, b = 4, f(n) = \sqrt{n}, n^{\log_b a} = n^{\log_4 2} = n^{\frac{1}{2}}$
case-2 applies, Since,
$f(n) = \Theta(n^{\log_b a}) = \Theta(n^{\frac{1}{2}})$ Thus, solution to the recurrence is,

$$T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(\sqrt{n} \lg n)$$

g)

$T(n) = T(n-2) + n^2$
By applying counting method here, At 0^{th} step, the recurrence result is $\implies n^2$.
Therefore,
Step Result
$0^{\text{th}} \implies n^2$
$1^{\text{st}} \implies (n-2)^2$
$2^{\text{nd}} \implies (n-4)^2$
.
.
$k^{\text{th}} \implies (n - 2(k))^2 = 0$ Because, recurrence will over at that step. So the recurrence value should be 0 at k^{th} step.
Therefore, by solving above equation we will get $k = \frac{n}{2}$. So, the solution of the recurrence is,

$$T(n) = n^2 + n^2 + n^2 + ... + (k^{\text{th}} time)n^2 = (\frac{n}{2})\, times\, n^2$$

Since $k = \frac{n}{2}$.

$$T(n) = \Theta(\frac{n}{2} \cdot n^2)$$
$$T(n) = \Theta(n^3)$$

<u>Problem : 4.3</u>

a)

$T(n) = 4T(\frac{n}{3}) + n \lg n$
By master theorem,

$a = 4, b = 3, f(n) = n \lg n, n^{\log_b a} = n^{\log_3 4} \approx n^{1.26}$
Since, $f(n) = O(n^{\log_3 4 - \epsilon}), \epsilon \approx 0.26$
We can apply case-1,

$$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_3 4})$$

b)

$T(n) = 3T(n/3) + \frac{n}{\lg n}$
By applying master theorem,
$a = 3, b = 3, f(n) = n/\lg n, n^{\log_b a} = n$
Let's say $f(n) = n/\lg n = O(n)$
$\therefore$ case-1 applies,
Since $f(n) = n/\lg n$ is asymptotically less than $n^{\log_b a} = n$, So that $f(n) = O(n^{\log_b a} + \epsilon)$ will always true for some positive value of ϵ , $\epsilon > 0$
$\therefore$ According to case-1 ,

$$T(n) = \Theta(n^{\log_b a}) = n$$

c)

$T(n) = 4T(n/4) + n^2 \cdot \sqrt{n}$
$\therefore a = 4, b = 2, f(n) = n^{5/2}, n^{\log_b a} = n^2$
case-3 of master theorem applies, $T(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon = 0.5$
Case-3 applies, Let's prove regularity condition for $f(n)$, for sufficient large value of n and $c < 1$.
$af(\frac{n}{b}) = 4 \cdot (\frac{n}{2})^{\frac{5}{2}} = \frac{4}{2^{5/2}} \cdot n^{5/2} \leq c \cdot f(n)$
As $f(n) = n^{5/2}$, We can take $c = \frac{1}{\sqrt{2}}$ which is less than 1.
So, by case-3 the solution to the recurrence is,

$$T(n) = \Theta(f(n)) = \Theta(n^2 \cdot \sqrt{n})$$

d)

$T(n) = 3T(\frac{n}{3} - 2) + \frac{n}{2}$
This recurrence is not in desired form of,
$T(n) = aT(n/b) + f(n)$
$\therefore$ Master theorem cannot apply.

e)

$T(n) = 2T(n/2) + n/\lg n$ By applying master theorem,
$a = 2, b = 2, f(n) = n/\lg n, n^{\log_b a} = n$
Let's say $f(n) = n/\lg n = O(n)$ (Same as problem 4.3 b)
$\therefore$ case-1 applies,
Since $f(n) = n/\lg n$ is asymptotically less than $n^{\log_b a} = n$, So that $f(n) = O(n^{\log_b a} + \epsilon)$ will always true for some positive value of ϵ , $\epsilon > 0$
$\therefore$ According to case-1 ,

$$T(n) = \Theta(n^{\log_b a}) = n$$

f)

$T(n) = T(n/2) + T(n/4) + T(n/8) + n$
This recurrence is not in desired form of,

$T(n) = aT(n/b) + f(n)$
$\therefore$ Master theorem cannot apply.

g)

$T(n) = T(n-1) + 1/n$ By applying counting method here, At 0^{th} step, the recurrence result is $\implies 1/n$.
Therefore,
Step Result
$0^{\text{th}} \implies 1/n$
$1^{\text{st}} \implies 1/(n-1)$
$2^{\text{nd}} \implies 1/(n-2)$
.
.
$k^{\text{th}} \implies 1/(n-k)$ The value of $n-k = 1$ because, denominator cannot be 0. So the recurrence value should be 1 at k^{th} step.
$\therefore$ By solving above equation we will get $n = k - 1$.

$$\therefore T(n) = \sum_{k=1}^{n+1} \frac{1}{k} = \lg n$$

$$\therefore T(n) = \Theta(\lg n)$$

h)

$T(n) = T(n-1) + \lg n$
Our recurrence not in the form of

$$T(n) = aT(n/b) + f(n)$$

But we can try another method like counting..
At 0^{th} step our recurrence value will be $\lg n$.
Therefore,
Step Result
$1^{\text{th}} \implies \lg n$
$2^{\text{st}} \implies \lg(n-1)$
$3^{\text{nd}} \implies \lg(n-2)$
.
.
$k^{\text{th}} \implies \lg(n-k-1)$
$\therefore n = k + 1$
So the solution to the recurrence is,

$$T(n) = (\lg n + \lg(n-1) + ... + \lg(n-k-1) = \sum_{k=1}^{n} \lg k = n \lg n$$

$$\therefore T(n) = \Theta(n \cdot \lg n)$$

i)

$T(n) = T(n-2) + \frac{1}{\lg n}$ Our recurrence not in the form of

$$T(n) = aT(n/b) + f(n)$$

But we can try another method like counting..
Therefore,
Step Result
$0^{\text{th}} \implies (n-0)$
$1^{\text{st}} \implies (n-2)$
$2^{\text{nd}} \implies (n-4)$
.
.
$(k)^{\text{th}} \implies (n-2k) = 0 \implies k = \frac{n}{2}$, So

$$T(n) = \frac{n}{2}\, times\, \frac{1}{\lg n}$$

$$\therefore T(n) = \Theta(\frac{n}{2} \cdot \frac{1}{\lg n}) = \Theta(\frac{n}{\lg n})$$

j)

$T(n) = \sqrt{n}T(\sqrt{n}) + n$
This recurrence should be in the form of,
$T(n) = aT(n/b) + f(n)$ where $a \geq 1, b > 1$ are constant And $f(n)$ is a function.
In given recurrence $a = \sqrt{n}$ is not a constant.
$\therefore$ Master theorem cannot apply.