Homework 3

蘇俊杰、劉韋成、曾守瑜

Question1

(3.4) Below is a theorem from Manber's book:

> For all constants $c > 0$ and $a > 1$, and for all monotonically increasing functions $f(n)$, we have $(f(n))^c = o(a^{f(n)})$.

Prove, by using the above theorem, that for all constants $a, b > 0$, $(\log_2 n)^a = o(n^b)$.

Question1

$(f(n))^c = o(a^{f(n)})$
Use $\log_2 n$ to replace $f(n)$.
$\{\log_2 n$ is a monotonically increasing function$\}$
Use a to replace c. $\{a > 0\}$
Use 2^b to replace a. $\{b > 0, 2^b > 1\}$

$(\log_2 n)^a = o((2^b)^{\log_2 n})$
$\Rightarrow (\log_2 n)^a = o(n^{b \ \log_2 2})$
$\Rightarrow (\log_2 n)^a = o(n^b)$

Question2

(3.5) For each of the following pairs of functions, say whether $f(n) = O(g(n))$ and/or $f(n) = \Omega(g(n))$. Justify your answers.

	$f(n)$	$g(n)$
(a)	$(\log n)^{\log n}$	$\frac{n}{\log n}$
(b)	$n^3 2^n$	3^n

Question2

(big-o) $f(n) = O(g(n))$:
$\exists c$, you can find an N such that $\forall n \geq N$, $f(n) \leq c \cdot g(n)$ holds.

(little-o) $f(n) = o(g(n))$:
$\forall c$, you can find an N such that $\forall n \geq N$, $f(n) \leq c \cdot g(n)$ holds.

So we can use $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$ to verify $f(n) = o(g(n))$.

Because $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$ means that the growth of $g(n)$ is obviously larger than the one of $f(n)$, which means that if n is large enough, $f(n) \leq c \cdot g(n)$ must hold, the constant c cannot influence this inequality.

Question2

(big-o) $f(n) = O(g(n))$:
$\exists c$, you can find an N such that $\forall n \geq N$, $f(n) \leq c \cdot g(n)$ holds.

(little-o) $f(n) = o(g(n))$:
$\forall c$, you can find an N such that $\forall n \geq N$, $f(n) \leq c \cdot g(n)$ holds.

By definition, we can see that $f(n) = o(g(n))$ implies $f(n) = O(g(n))$.

Question2

(big-omega) $f(n) = \Omega(g(n))$:
$\exists c$, you can find an N such that $\forall n \geq N$, $f(n) \geq c \cdot g(n)$ holds.

(little-omega) $f(n) = \omega(g(n))$:
$\forall c$, you can find an N such that $\forall n \geq N$, $f(n) \geq c \cdot g(n)$ holds.

So similarly, we can use $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$ to verify $f(n) = \omega(g(n))$.

We can also find that $[f(n) = \omega(g(n))] = [g(n) = o(f(n))]$ because they are all $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

Question2

(big-omega) $f(n) = \Omega(g(n))$:
$\exists c$, you can find an N such that $\forall n \geq N$, $f(n) \geq c \cdot g(n)$ holds.

(little-omega) $f(n) = \omega(g(n))$:
$\forall c$, you can find an N such that $\forall n \geq N$, $f(n) \geq c \cdot g(n)$ holds.

By definition, we can see that $f(n) = \omega(g(n))$ implies $f(n) = \Omega(g(n))$.

Question2

(little-o) $f(n) = o(g(n))$:
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

(big-omega) $f(n) = \Omega(g(n))$:
$\exists c$, you can find an N such that $\forall n \geq N$, $f(n) \geq c \cdot g(n)$ holds.

By definition, we can see that $f(n) = o(g(n))$ implies $f(n) \neq \Omega(g(n))$ because the growth of $g(n)$ is obviously larger than the one of $f(n)$, which means that if n is large enough, $c \cdot g(n)$ must be larger than $f(n)$, and $f(n) \geq c \cdot g(n)$ will never hold at that moment, no matter how small the constant c is.

Similarly, $f(n) = \omega(g(n))$ implies $f(n) \neq O(g(n))$.

Question2(a)

$f(n) = (\log n)^{\log n}$, $g(n) = \frac{n}{\log n}$

[Claim] $f(n) = \Omega(g(n)) \wedge f(n) \neq O(g(n))$

To prove the claim, we just need to prove that $g(n) = o(f(n))$,

which is, $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

Because $g(n) = o(f(n))$ implies $f(n) = \Omega(g(n))$, and

$g(n) = o(f(n))$ implies $f(n) \neq O(g(n))$.

Question2(a)

$$
\begin{aligned}
\lim_{n\to\infty} \frac{g(n)}{f(n)} &= \lim_{n\to\infty} \frac{\frac{n}{\log n}}{(\log n)^{\log n}} \\
&= \lim_{n\to\infty} \frac{n}{\log n \cdot (\log n)^{\log n}} \\
& \{\text{use L'Hôpital's rule}\} \\
&= \lim_{n\to\infty} \frac{1}{\frac{(\log n)^{\log n} \cdot ((\ln 2)(\log n)\log(\log n) + (\log n) + 1)}{n(\ln 2)}} \\
&= 0
\end{aligned}
$$

Hence, $f(n) = \Omega(g(n))$ and $f(n) \neq O(g(n))$.

Question2(a)

$\mathrm{f(n)}= (\log n)^{\log n}$,$\mathrm{g(n)}= \frac{n}{\log n}$

Claim $\mathrm{f(n)}=\Omega(\mathrm{g(n)})$, $\exists$ c $\exists$ N $\forall$ n $\geq$N, let $c = 1$

$(logn)^{logn} \geq c \cdot (\frac{n}{logn})$

$\xLeftarrow{logn \text{ 以 } x \text{ 代入}} x^x \geq c(\frac{2^x}{x})$

$\xLeftarrow{\text{兩邊同取 } log_2} xlogx \geq x + logc - logx$ (當 x>1 成立)

When n>2,c=1

f(n)=Ω(g(n))

Question2(b)

$f(n) = n^3 2^n$, $g(n) = 3^n$

[Claim] $f(n) = O(g(n)) \wedge f(n) \neq \Omega(g(n))$

To prove the claim, we just need to prove that $f(n) = o(g(n))$,

which is, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

Because $f(n) = o(g(n))$ implies $f(n) = O(g(n))$, and

$$f(n) = o(g(n)) \text{ implies } f(n) \neq \Omega(g(n)).$$

Question2(b)

$$
\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^3 2^n}{3^n} &= \lim_{n\to\infty} \frac{n^3}{(\frac{3}{2})^n} \\
&\{\text{use L'Hôpital's rule 3 times}\} \\
&= \lim_{n\to\infty} \frac{3n^2}{(\frac{3}{2})^n(\ln\frac{3}{2})} \\
&= \lim_{n\to\infty} \frac{6n}{(\frac{3}{2})^n(\ln\frac{3}{2})^2} \\
&= \lim_{n\to\infty} \frac{6}{(\frac{3}{2})^n(\ln\frac{3}{2})^3} \\
&= 0
\end{aligned}
$$

Hence, $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$.

Question3

Suppose $f(n)$ is a strictly increasing function, i.e., if $n_1 < n_2$, then $f(n_1) < f(n_2)$, and $f(n) = O(g(n))$. Is it true that $\log f(n) = O(\log g(n))$? Please justify your answer. What about $2^{f(n)} = O(2^{g(n)})$? What if $f(n)$ is constant?

Question3

這一題有四個問題

每個問題都要解釋加回答

嚴格遞增兩題 15 分 constant 兩題 5 分

Question3(1)

$log(f(n)) = O(log(g(n)))$

根據 $f(n) = O(g(n))$ 的定義可得:

there exist constants c and N
such that, for all $n \geq N$, $f(n) \leq c \times g(n)$

{對於 $g(n) \leq cf(n)$ 兩邊同取 log}

$log(f(n)) \leq log(c) + log(g(n))$

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

$\frac{log(f(n))}{log(g(n))} - 1 \leq \frac{log(c)}{log(g(n))}$

Question3(1) cont.

$$\frac{log(f(n))}{log(g(n))} \leq \frac{log(c)}{log(g(n))} + 1$$

$$log(f(n)) \leq (\frac{log(c)}{log(g(n))} + 1) \times log(g(n))$$

{根據開頭所述：for all $n \geq N$}

$$log(f(n)) \leq (\frac{log(c)}{log(g(N))} + 1) \times log(g(n))$$

{set $c' = \frac{log(c)}{log(g(N))} + 1$ (c' is constant)}

$$log(f(n)) \leq c' \times log(g(n))$$

$log(f(n)) = O(log(g(n)))$ is true.

Question3(2)

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

可以找到反例：

set $f(n) = 2log_2(n)$ and $g(n) = log_2(n)$
其滿足條件：$2log_2(n) = O(log_2(n))$

將其帶入式子可以得到：

$2^{2log_2(n)} = O(2^{log_2(n)})$
$n^2 = O(n)$，發生矛盾！(因為 $n^2 \neq O(n)$)

所以 $2^{f(n)} = O(2^{g(n)})$ is false.

Question3(3)

$log(f(n)) = O(log(g(n)))$ constant case

因為 $f(n)$ 為 constant, $log(f(n))$ 亦為 constant

又根據定義：there exist constants c and N
such that, for all $n \geq N$, $f(n) \leq c \times g(n)$

set $g(n) = 1$, 代表 $log(g(n)) = 0$

$log(f(n)) \leq c \times log(g(n)) = c \times 0 = 0$

$\{$舉例：$log(2) \leq c \times log(g(n)) = c \times 0 = 0$ 矛盾$\}$

並沒有滿足條件

所以 $log(f(n)) = O(log(g(n)))$ is false.

Question3(4)

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

因為 $f(n)$ 為 constant

又根據定義：there exist constants c and N such that, for all $n \geq N$, $f(n) \leq c \times g(n)$

set $g(n) = 1$, 代表 $2^{g(n)} = 2$

$\exists c > 0$, s.t. $2^{f(n)} \leq c \times 2^{g(n)} = c \times 2$

所以 $2^{f(n)} = O(2^{g(n)})$ is true.

Question4

(3.12) Solve the following recurrence relation:

$$\begin{cases} T(1) = 1 \\ T(n) = n + \sum_{i=1}^{n-1} T(i), \quad n \geq 2 \end{cases}$$

Question4

$T(1)=1$

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

$T(3)=3+T(2)+T(1)$

$\vdots$

$T(n\text{-}1)=(n\text{-}1)+[T(n\text{-}2)+T(n\text{-}3)+\ldots]\ \ldots$ ①

$T(n)=(n)+[T(n\text{-}1)+T(n\text{-}2)+T(n\text{-}3)+\ldots]\ldots$ ②

Question4

$T(n-1)=(n-1)+[T(n-2)+T(n-3)+\ldots]\ \ldots$ ①

$T(n)=(n)+[T(n-1)+T(n-2)+T(n-3)+\ldots]\ldots$ ②

② − ① :

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

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

Question4

$$
\begin{aligned}
\text{T(n)} &= \text{2T(n-1)}+1 \\
&= \text{2[2T(n-2)+1]}+1 \\
&= 2^2\text{T(n-2)}+(2+1) \\
&= 2^2\text{[2T(n-3)+1]}+(2+1) \\
&= 2^3\text{T(n-3)}+(2^2+2+1) \\
&\vdots \\
&= 2^i\text{T(n-i)}+(1+2+2^2+\ldots+2^{i-1})
\end{aligned}
$$

Question4

2^iT(n-i)+$(1+2+2^2+...+2^{i-1})$

i 以 (n-1) 代入:

$=2^{n-1}\cdot$T(1)+$(1+2+...+2^{n-2})$

$=2^{n-1}\cdot 1+\frac{2(2^{n-1}-1)}{2}$

$=2^n$-1

Question5

(3.18) Consider the recurrence relation

$$T(n) = 2T(n/2) + 1,\ T(2) = 1.$$

We try to prove that $T(n) = O(n)$ (we limit our attention to powers of 2). We guess that $T(n) \leq cn$ for some (as yet unknown) c, and substitute cn in the expression. We have to show that $cn \geq 2c(n/2) + 1$. But this is clearly not true. Find the correct solution of this recurrence (you can assume that n is a power of 2), and explain why this attempt failed.

Question5

$$T(n) = 2T(n/2) + 1, \; T(2) = 1$$

這題可以點出題幹證明錯在哪，並修正題幹的證明
也可以直接找出 T 的通解再去解釋題幹證明錯在哪

Question5

那麼為何題目一開始的證明會出問題呢？

$$T(n) \leq cn$$

這個式子是對的，接下來依照 $T(n)$ 的定義代換

$$2T(n/2) + 1 \leq cn$$

問題就在於，得到這式子之後，題目又自動將 $T(n/2)$ 換成 $c(n/2)$。但是

$$T(n/2) \leq c(n/2)$$
$$2T(n/2) + 1 \leq 2c(n/2) + 1$$

我們無從得知 $2c(n/2) + 1$ 與 cn 之間的大小關係，我們只知道它們都比 $2T(n/2) + 1$ 來得大。所以題目給出的不等式是有問題的。

Question5

那麼要怎麼修正呢？
縮緊條件，改成證明 $T(n) \leq cn - 1$
因為證明這件事就隱含 $T(n) \leq cn$
Base case: $\exists c = 1$ s.t. $T(2) = 2 - 1 = 1$
Inductive: $T(n) = 2T(n/2) + 1$
By inductive hypothesis, $T(n/2) \leq c(n/2) - 1$
So $2T(n/2) + 1 \leq 2(c(n/2) - 1) + 1 = cn - 1$
Therefore, $T(n) = 2T(n/2) + 1 \leq cn - 1$

Question5

大家都知道要改證明 $T(n) \leq cn - 1$
但要注意的是，必須寫出 Base case。雖然題目在秀出錯誤證明時沒有提到 Base case，但如果真的沒有 Base case，那麼 $T(n/2)$ 的存在就會在 $n = 2$ 時失去意義。
就算假設題目已知 $T(2) \leq cn$，但並無法直接說 $T(2) \leq cn - 1$ 也會成立。

Question5

$$T(n) = 2T(n/2) + 1, T(2) = 1$$

另一種求解方法是用其他方法證明 $T(n) = O(n)$，但別忘了點出題幹錯在哪

T 滿足 $T(n) = aT(n/b) + O(n^k)$ 的形式，其中 $a = 2, b = 2, k = 0$

已知這類 recurrence relations 的解為：

$$T(n) = \begin{cases} O(n^{\log_b a}) & \text{if } a > b^k \\ O(n^k \log n) & \text{if } a = b^k \\ O(n^k) & \text{if } a < b^k \end{cases}$$

$a = 2 > 1 = b^k$，於是 $T(n) = O(n^{\log_2 2}) = O(n)$

Question5

也可以直接找出 T 的通式
觀察 T 的規律

$$\begin{aligned} T(2) &= 1 \\ T(4) &= 2 \times 1 + 1 = 3 \\ T(8) &= 2 \times 3 + 1 = 7 \\ T(16) &= 2 \times 7 + 1 = 15 \\ &\vdots \end{aligned}$$

猜測 $T(n) = n - 1$，並用歸納法證明
也能看出 $T(n) = O(n)$