Homework 3

Yu-Ju Teng Ling-Hsuan Chen

Question1

1. (3.5) For each of the following pairs of functions, determine whether $f(n) = O(g(n))$ and/or $f(n) = \Omega(g(n))$. Justify your answers.

	$f(n)$	$g(n)$
(a)	$\sqrt{n}$	$(\log n)^2$
(b)	$n^4 2^n$	4^n

Question1

(big-o) $f(n) = O(g(n))$:
$\exists c\ , N > 0\ s.t.\ f(n) \leq c \cdot g(n)$ holds $\forall n \geq N$.

(big-omega) $f(n) = \Omega(g(n))$:
$\exists c\ , N > 0\ s.t.\ f(n) \geq c \cdot g(n)$ holds $\forall n \geq N$.

(little-o) if $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$, then $f(n) = o(g(n))$

$\Rightarrow$ if $f(n) = o(g(n))$, then $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$.

$\Rightarrow$ if $g(n) = o(f(n))$, then $f(n) = \Omega(g(n))$ and $f(n) \neq O(g(n))$.

Question1 (a)

$f(n) = \sqrt{n}$, $g(n) = (logn)^2$:

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{(logn)^2}{\sqrt{n}}$$

$$\{\text{use L'Hôpital's rule}\}$$

$$= \lim_{n\to\infty} \frac{\frac{2logn}{(ln10)n}}{\frac{1}{2\sqrt{n}}} = \lim_{n\to\infty} \frac{4logn}{ln10\sqrt{n}}$$

$$\{\text{use L'Hôpital's rule}\}$$

$$= \lim_{n\to\infty} \frac{\frac{4}{(ln10)n}}{\frac{ln10}{2\sqrt{n}}} = \lim_{n\to\infty} \frac{8}{(ln10)^2\sqrt{n}}$$

$$= 0$$

Question1 (a)

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

which implies $f(n) = \Omega(g(n))$ and $f(n) \neq O(g(n))$.

If you lost the points because of omitting $ln 10$, you can get the point back this time.

Reminder: If you are not sure what the base of log is, you can assume what it is and write down your assumption.

Question1 (b)

$f(n) = n^4 2^n$, $g(n) = 4^n$:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^4 2^n}{4^n} = \lim_{n\to\infty} \frac{n^4}{2^n}$$

$$\{\text{use L'Hôpital's rule 4 times}\}$$

$$= \lim_{n\to\infty} \frac{24}{(ln2)^4 2^n}$$

$$= 0$$

Question1 (b)

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

which implies $f(n) = O(g(n))$ and $f(n) \neq \Omega(g(n))$.

Question2

2. Suppose f is a strictly increasing function that maps every positive integer to another positive integer, i.e., if $1 \leq n_1 < n_2$, then $1 \leq f(n_1) < f(n_2)$, and $f(n) = O(g(n))$ for some other function g. Is it true that $\log f(n) = O(\log g(n))$? Please justify your answer. How about $2^{f(n)} = O(2^{g(n)})$? Is it true?

Question2

(big-o) $f(n) = O(g(n))$:
$\exists c, N > 0$ $s.t.$ $f(n) \leq c \cdot g(n)$ holds $\forall n \geq N$.

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

$$\begin{aligned} log(f(n)) \leq log(c \cdot g(n)) &= log(c) + log(g(n)) \\ &= log(g(n)) \cdot (1 + \frac{log(c)}{log(g(n))}) \\ &\leq log(g(n)) \cdot (1 + \frac{log(c)}{log(g(N))}) \\ &\leq log(g(n)) \cdot c', \text{ where } c' > 0. \end{aligned}$$

$s.t.$ $log(f(n)) \leq c' \cdot log(g(n))$ holds $\forall n \geq N$.

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

Question2

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

We can only get $2^{f(n)} \leq 2^{c \cdot g(n)} = (2^{g(n)})^c$

and it doesn't imply $2^{f(n)} = O(2^{g(n)})$.

The hypothesis $2^{f(n)} = O(2^{g(n)})$ can simply be rejected with

a counter example: $f(n) = 2log_2(n)$, $g(n) = log_2(n)$.

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

$2^{f(n)} = 2^{2log_2(n)} = n^2 = O(n^2)$

$2^{g(n)} = 2^{log_2(n)} = n = O(n)$

$O(n^2) \notin O(n)$

$\Rightarrow 2^{f(n)} \neq O(2^{g(n)})$

Question3

3. (3.18) Consider the recurrence relation

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

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.

Question3

Find the correct solution of the recurrence:

n	2	4	8	16	$\cdots$	k	$\cdots$
$T(n)$	1	3	7	15	$\cdots$	$k-1$	$\cdots$

Assumption: $T(n) = n - 1$.

Question3

[Inductive hypothesis]: $T(n) = n - 1.$

[Base case]$(n = 2)$: $T(2) = 1 = 2 - 1 = n - 1.$

[Inductive step]$(n \geq 2)$:

$$\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + 1 \\ &= 2(\tfrac{n}{2} - 1) + 1 \ \{\text{by } I.H.\} \\ &= n - 1 \end{aligned}$$

Question3

Why the attempt of the question failed?
$cn \geq 2c(n/2) + 1$

By $T(n) = O(n)$, there exists some c such that:

$cn \geq T(n) = 2T(n/2) + 1$ Correct

$cn \geq T(n) = 2c(n/2) + 1$?

It replaces $2T(n/2)$ with $2c(n/2)$, but we only know that:

- $cn \geq T(n)$
- $2c(n/2) + 1 \geq 2T(n/2) + 1 = T(n)$

$\Rightarrow cn \geq T(n) = 2c(n/2) + 1$ Uncertain

Question3

By induction:

Proof: $T(n) \leq cn$ for some c

[Base case] $(n = 2)$: Let $c = 1$, $T(2) = 1 \leq 1 \times 2 = cn$.

[Inductive hypothesis]: $T(n) \leq c(n)$.

[Inductive step] $(n \geq 2)$:
$$\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + 1 \\ &\leq 2c(\tfrac{n}{2}) + 1 \text{ \{by } I.H.\text{\}} \\ &= cn + 1 \end{aligned}$$

Fail to prove that $T(n) \leq cn$ for some c. (There is an extra 1.)

Question3

So, we can fix it as follows:

Proof: $T(n) \leq cn$ - 1 for some c

[Base case]($n = 2$): Let $c = 1$, $T(2) = 1 \leq 1 \times 2 - 1 = cn$.

[Inductive hypothesis]: $T(n) \leq c(n) - 1$.

[Inductive step] ($n \geq 2$):
$$\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + 1 \\ &\leq 2(c(\tfrac{n}{2}) - 1) + 1 \ \{\text{by } I.H.\} \\ &= cn - 1 \end{aligned}$$

Since $cn - 1 \leq cn$, we can say that $T(n) \leq cn$ for some c.

Question4

4. Solve the following recurrence relation using *generating functions*. This is a very simple recurrence relation, but for the purpose of practicing you must use generating functions in your solution.

$$\begin{cases} T(0) = 0 \\ T(1) = 1 \\ T(n) = T(n-1) + 2\,T(n-2), \quad n \geq 2 \end{cases}$$

Question4

Let $T_n = T(n)$, $G(z) = T_0 + T_1 z + T_2 z^2 + \cdots + T_n z^n + \cdots$.

$$\begin{aligned} G(z) &= T_0 + T_1 z + T_2 z^2 + \cdots + T_n z^n + \cdots \\ zG(z) &= T_0 z + T_1 z^2 + T_2 z^3 + \cdots + T_{n-1} z^n + \cdots \\ 2z^2 G(z) &= 2T_0 z^2 + 2T_1 z^3 + 2T_2 z^4 + \cdots + 2T_{n-2} z^n + \cdots \\ \hline (1 - z - 2z^2) G(z) &= T_0 + T_1 z - T_0 z \\ &= 0 + z - 0 \\ &= z \end{aligned}$$

$$G(z) = \frac{z}{1 - z - 2z^2}$$

Question4

$$
\begin{aligned}
G(z) &= \frac{z}{1 - z - 2z^2} \\
&= \frac{z}{(1+z)(1-2z)} \\
&= \frac{\frac{-1}{3}}{1+z} + \frac{\frac{1}{3}}{1-2z} \\
&= T_0 + T_1 z + T_2 z^2 + \cdots + [\frac{-1}{3}(-1)^n + \frac{1}{3}(2^n)]z^n + \cdots
\end{aligned}
$$

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

Question5

5. The Fibonacci word sequence of bit strings is defined as follows:

$$FW(i) = \begin{cases} 0 & \text{if } i = 0 \\ 1 & \text{if } i = 1 \\ FW(i-1) \cdot FW(i-2) & \text{if } i \geq 2 \end{cases}$$

Here $\cdot$ denotes the operation of string concatenation. The first six Fibonacci words (from $FW(0)$ to $FW(5)$), for instance, are: 0, 1, 10, 101, 10110, 10110101. A given bit pattern may or may not occur in a Fibonacci word (as a substring). For instance, 10101 occurs in $FW(5)$ (but not $FW(4)$ or earlier), while 11101 never occurs in any Fibonacci word.

Suppose we are given a bit pattern p (that is not just a single bit) and asked to determine whether it occurs in some Fibonacci word. To be efficient, we certainly want to search as few Fibonacci words as possible for the pattern. Please give a lower bound m_l and an upper bound m_u in terms of $F^{-1}(|p|)$ ($|p|$ denotes the length of p), as tight as possible, such that, if p will ever occur in any Fibonacci word, it occurs already in some word between $FW(m_l)$ and $FW(m_u)$ inclusively.

Note: Let $F^{-1}(n)$ (for $n \geq 2$) denote the index value i ($i \geq 2$) such that $F(i-1) < n \leq F(i)$, i.e., n is between the two Fibonacci numbers $F(i-1)$ and $F(i)$, possibly equal to $F(i)$ but greater than $F(i-1)$; here, the Fibonacci numbers start from 1, which is indexed as the 0-th element in the sequence. For instance, $F^{-1}(2) = 2$, $F^{-1}(3) = 3$, $F^{-1}(5) = 4$, $F^{-1}(8) = 5$, and $F^{-1}(6) = F^{-1}(7) = 5$.

Question5

For m_l, we want to try the shortest word which has at least $|p|$ bit.

$\Rightarrow m_l = F^{-1}(|p|)$

Some people claim that m_l should be tighter because if it can be found in $F^{-1}(|p|)$, it can be found in the words longer than $F^{-1}(|p|)$ as well.

However, in reality, we have to compose words step by step. For example, we get $F^{-1}(|p|) + 2$ by concatenating $F^{-1}(|p|) + 1$ and $F^{-1}(|p|)$. Therefore, if we can find pattern p in $F^{-1}(|p|)$, we don't need to make an effort to compose more words.

If you wrote the answer bigger than $F^{-1}(|p|)$ with proper explanation, you can get the points.

Question5

For m_u, we try to find pattern p between substrings.

Let $FW(F^{-1}(|p|)) = s_0$, $FW(F^{-1}(|p|) + 1) = s_1$, ...

$FW(F^{-1}(|p|) - 1) = s_{-1}$, $FW(F^{-1}(|p|) - 2) = s_{-2}$, ...

If we cannot find pattern p in s_0, we can try s_1 to check if there is any new pattern whose length is longer than $|p|$.

$FW(F^{-1}(|p|)) = s_0 = s_{-1} \cdot s_{-2}$

Since the length of $s_0 \geq |p|$, the length of $s_{-1} + s_{-2} \geq |p|$.

$FW(F^{-1}(|p|) + 1) = s_1 = s_0 \cdot s_{-1} = s_{-1} \cdot s_{-2} \cdot s_{-1}$

Question5

Now we try s_2:

$FW(F^{-1}(|p|) + 2) = s_2 = s_1 \cdot s_0 = s_{-1} \cdot s_{-2} \cdot s_{-1} \cdot s_{-1} \cdot s_{-2}$

Since the length of $s_{-1} > s_{-2}$, the length of $s_{-1} + s_{-1} > |p|$.

Now we try s_3:

$FW(F^{-1}(|p|)+3) = s_3 = s_2 \cdot s_1 = s_{-1} \cdot s_{-2} \cdot s_{-1} \cdot s_{-1} \cdot s_{-2} \cdot s_{-1} \cdot s_{-2} \cdot s_{-1}$

We cannot find any new pattern here!

We cannot find any new pattern in other words neither!
(For example, it is impossible to see $s_{-2} \cdot s_{-2} \cdot s_{-2}$.)

$\Rightarrow m_u = F^{-1}(|p|) + 2$