CS170–Spring 2019 — Homework 2 Solutions

Ran Liao, SID 3034504227

Collaborators: YiLin Wu, YeFan Zhou

1. Study Group

Name	SID
Ran Liao	3034504227
YiLin Wu	3034503863
YeFan Zhou	3034503824

2. Asymptotic Complexity Comparisons

(a) $3 \le 7 \le 2 \le 5 \le 4 \le 9 \le 8 \le 6 \le 1$

(b) (i) $\log_3 n = \Theta(\log_4 n)$

Proof:

$$\log_3 n = \log_{(4^{\log_4 3})} n = (\frac{1}{\log_4 3}) \log_4 n = \Theta(\log_4 n)$$

(ii) $n \log(n^4) = O(n^2 \log(n^3))$

Proof:

$$\lim_{n\to+\infty} \frac{n^2 \log(n^3)}{n \log(n^4)} = \lim_{n\to+\infty} \frac{3n^2 \log n}{4n \log n} = \lim_{n\to+\infty} \frac{3}{4} n = \infty$$

(iii) $\sqrt{n} = \Omega((\log n)^3)$

Proof: (Use L'Hôpital's rule)

$$\begin{aligned}\lim_{n\to+\infty} \frac{\sqrt{n}}{(\log n)^3} &= \lim_{n\to+\infty} \frac{\frac{1}{2\sqrt{n}}}{3(\log n)^2 \frac{1}{n}} = \lim_{n\to+\infty} \frac{\sqrt{n}}{(\log n)^2} \\ &= \lim_{n\to+\infty} \frac{\frac{1}{2\sqrt{n}}}{2(\log n) \frac{1}{n}} = \lim_{n\to+\infty} \frac{\sqrt{n}}{\log n} \\ &= \lim_{n\to+\infty} \frac{\frac{1}{2\sqrt{n}}}{\frac{1}{n}} = \lim_{n\to+\infty} \sqrt{n} = \infty\end{aligned}$$

(iv) $2^n = \Theta(2^{n+1})$

Proof:

$$2^{n+1} = 2 * 2^n = \Theta(2^n)$$

(v) $n = \Omega((\log n)^{\log\log n})$

Proof:(Use L'Hôpital's rule)

Take the logarithm of both functions

$$\log(\log n)^{\log\log n} = (\log\log n)^2$$

Therefore, it is equivalent to compare $(\log\log n)^2$ and $\log n$

$$\begin{aligned}\lim_{n\to+\infty} \frac{(\log\log n)^2}{\log n} &= \lim_{n\to+\infty} \frac{2(\log\log n)\frac{1}{\log n}\frac{1}{n}}{\frac{1}{n}} = \lim_{n\to+\infty} \frac{2(\log\log n)}{\log n} \\ &= \lim_{n\to+\infty} \frac{\frac{1}{\log n}\frac{1}{n}}{\frac{1}{n}} = \lim_{n\to+\infty} \frac{1}{\log n} = 0\end{aligned}$$

(vi) $n + \log n = \Theta(n + (\log n)^2)$

Proof:(Use L'Hôpital's rule)

$$\begin{aligned}\lim_{n\to+\infty} \frac{n + \log n}{n + (\log n)^2} &= \lim_{n\to+\infty} \frac{1 + \frac{1}{n}}{1 + 2(\log n)\frac{1}{n}} = \lim_{n\to+\infty} \frac{n+1}{n + 2\log n} \\ &= \lim_{n\to+\infty} \frac{1}{1 + \frac{2}{n}} = 1\end{aligned}$$

(vii) $\log(n!) = O(n \log n)$

Proof: Since $n \log n = log(n^n)$, it's equivalent to compare $n!$ and n^n. Obviously, $n! = O(n^n)$.

3. In Between Functions

4. Bit Counter

n	# Total Flips
1	1
2	4
3	11
4	26

There're two stages when counting from 0 to $2^n - 1$ for a n-bit counter.

- Stage1 : when most-significant bit is 0
- Stage2 : when most-significant bit is 1

Suppose $f(n)$ represents the total number of flips need for n-bit long counter. Both in stage 1 and 2, $f(n-1)$ flips is needed to flip all bits to 1 except the most-significant bit. When switching between stage 1 and 2, another n flips is introduced. 1 flip to switch the most-significant bit from 0 to 1. $n-1$ flips needed to switch all other bits to 0. Therefore, we have the following recursive formula:

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

To solve this formula, we keep using it recursively, providing $f(1) = 1$ as base case:

$$\begin{aligned}
f(n) &= 2f(n-1) + n \\
&= 4f(n-2) + 2(n-1) + n \\
&= 8f(n-3) + 4(n-2) + 2(n-1) + n \\
&= 2^k f(n-k) + 2^{k-1}(n-k+1) + \ldots + 2^2(n-2) + 2^1(n-1) + 2^0(n-0) \\
&= 2^{n-1} f(1) + 2^{n-2}(2) + \ldots + 2^2(n-2) + 2^1(n-1) + 2^0(n-0) \\
&= 2^{n-1}(1) + 2^{n-2}(2) + \ldots + 2^2(n-2) + 2^1(n-1) + 2^0(n-0) \qquad (1)
\end{aligned}$$

Then we multiply it by 2:

$$2f(n) = 2^n(1) + 2^{n-1}(2) + \ldots + 2^3(n-2) + 2^2(n-1) + 2^1(n-0) \qquad (2)$$

By (2) - (1), we have

$$\begin{aligned}
2f(n) - f(n) &= 2^n + 2^{n-1} + \ldots + 2^1 - n \\
f(n) &= \frac{2(1-2^n)}{1-2} - n \\
&= 2^{n+1} - n - 2 \\
&= \Theta(2^n)
\end{aligned}$$