CSCI 3104 Fall 2022
Prof. Grochow and Nagesh

Problem Set 5

Due Date .... Monday October 10, 2022 8pm MT
Name .... **Tyler Huynh**
Student ID **109603994**
Collaborators .... **List Your Collaborators Here**

Contents

Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Useful links and references on LaTeXcan be found here on Canvas.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding**

of the material. If there is any confusion about this policy, it is your responsibility to clarify before the due date.

- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section Honor Code). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign the Honor Pledge)

Problem HC. On my honor, my submission reflects the following:

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

In the specified region below, clearly indicate that you have upheld the Honor Code. Then type your name.

Honor Pledge. I, **Tyler Huynh** on my honor pledge that my submission is a reflection of my own understanding of the material, any and all collaborations/sources have been properly cited, I have not posted any material to external sources, and I have not copied other solutions as my own. □

12 Standard 12- Asymptotics I (Calculus I techniques)

Problem 12. For each part, you will be given a list of functions. Your goal is to order the functions from **slowest growing** to **fastest growing**. That is, if your answer is $f_1(n), \ldots, f_k(n)$, then it should be the case that $f_i(n) \leq O(f_{i+1}(n))$ for all i. If two adjacent functions have the same order of growth (that is, $f_i(n) = \Theta(f_{i+1}(n))$), clearly specify this. **Show all work, including Calculus details.** Plugging into WolframAlpha is not sufficient.

You may find the following helpful.

- Recall that our asymptotic relations are transitive. So if $f(n) \leq O(g(n))$ and $g(n) \leq O(h(n))$, then $f(n) \leq O(h(n))$. The same applies for little-o, Big-Theta, etc. Note that the goal is to order the growth rates, so transitivity is very helpful. We encourage you to make use of transitivity rather than comparing all possible pairs of functions, as using transitivity will make your life easier.
- You may also use the Limit Comparison Test. However, you **MUST** show all limit computations at the same level of detail as in Calculus I-II. Should you choose to use Calculus tools, whether you use them correctly will count towards your score.
- You may **NOT** use heuristic arguments, such as comparing degrees of polynomials or identifying the "high order term" in the function.
- If it is the case that $g(n) = c \cdot f(n)$ for some constant c, you may conclude that $f(n) = \Theta(g(n))$ without using Calculus tools. You must clearly identify the constant c (with any supporting work necessary to identify the constant- such as exponent or logarithm rules) and include a sentence to justify your reasoning.

You may also find it helpful to order the functions using an `itemize` block, with the work following the end of the `itemize` block.

- This function grows the slowest: $f_1(n)$
- These functions grow at the same asymptotic rate and faster than $f_1(n)$: $f_2(n), f_3(n), \ldots$
- These functions grow at the same asymptotic rate, but faster than $f_2(n)$: $f_k(n)$.

Also below is an example of an `align` block to help you organize your work.

$$\begin{aligned}
\lim_{n \to \infty} \frac{n^2}{2^n} &= \lim_{n \to \infty} \frac{2n}{\ln(2) \cdot 2^n} \\
&= \lim_{n \to \infty} \frac{2}{(\ln(2))^2 \cdot 2^n} \\
&= 0.
\end{aligned}$$

12.1 Problem 12(a)

(a) $n^2 + 200n, \qquad n + 1000000, \qquad n^3 - 20n^2, \qquad \frac{n^2}{\sqrt{n}}.$

Answer. Let us first define our values for further use:

$$\begin{aligned} f_1(n) &= n^2 + 200n \\ f_2(n) &= n + 1000000 \\ f_3(n) &= n^3 - 20n^2 \\ f_4(n) &= \tfrac{n^2}{\sqrt{n}} \end{aligned}$$

We will first start by comparing $f_1(n)$ with $f_2(n)$ to find which function will grow at a larger rate using L'hopitals rule:

$$\begin{aligned} \lim_{n\to\infty} \frac{n^2 + 200n}{n + 1000000} &= \lim_{n\to\infty} \frac{n^2 + 200n}{n + 1000000} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{2n + 200}{1} \\ &= \infty. \end{aligned}$$

We can see from the above that this will be $f_2(n) \leq O(f_1(n))$, which is representative that $f_1(n)$ will be an upper bound of $f_2(n)$. Thus, $f_1(n) \geq \Omega(f_2(n))$, which is the lower bound.

We will now compare $f_1(n)$ with $f_4(n)$ to find which function will grow at a larger rate using L'hopitals rule:

$$\begin{aligned} \lim_{n\to\infty} \frac{n^2 + 200n}{\frac{n^2}{\sqrt{n}}} &= \lim_{n\to\infty} \frac{n^2 + 200n}{n^{3/2}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{2n + 200}{\frac{3}{2}n^{\frac{1}{2}}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{2}{\frac{3}{4}n^{-\frac{1}{2}}} \\ &= \lim_{n\to\infty} \frac{2n^{-\frac{1}{2}}}{\frac{3}{4}} = \frac{\infty}{3} \\ &= \infty. \end{aligned}$$

We can see from the above that this will actually be $f_4(n) \leq O(f_1(n))$, which is representative that $f_1(n)$ will be an upper bound of $f_4(n)$. Thus, $f_1(n) \geq \Omega(f_4(n))$, which is the lower bound.

We will now compare $f_1(n)$ with $f_3(n)$ to find which function will grow at a larger rate using L'hopitals rule:

$$\begin{aligned} \lim_{n\to\infty} \frac{n^2 + 200n}{n^3 - 20n^2} &= \lim_{n\to\infty} \frac{n^2 + 200n}{n^3 - 20n^2} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{2n + 200}{3n^2 - 40n} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{2}{6n - 40} = \frac{2}{\infty} \\ &= 0. \end{aligned}$$

We can see from the above that $f_3(n)$ will actually grow faster than $f_1(n)$, such that $f_1(n) \leq O(f_3(n))$. This shows that $f_3(n)$ will be an upper bound of $f_1(n)$. Thus, $f_3(n) \geq \Omega(f_1(n))$, which is the lower bound. We can see that since $f_3(n)$ will grow faster than $f_1(n)$ and $f_1(n)$ grows faster than both $f_2(n)$ and $f_3(n)$, than we know that it will grow the fastest out of our set of functions.

We do not know whether $f_2(n)$ or $f_4(n)$ grows faster than each other so we will compare the two functions to each other using L'hopital's rule:

$$\begin{aligned}
\lim_{n\to\infty} \frac{n+1000000}{\frac{n^2}{\sqrt{n}}} &= \lim_{n\to\infty} \frac{n+1000000}{n^{3/2}} = \frac{\infty}{\infty} \qquad \mathit{Indeterminate} \\
&= \lim_{n\to\infty} \frac{1}{\frac{3}{2}n^{\frac{1}{2}}} = \frac{1}{\infty} \\
&= 0.
\end{aligned}$$

We can see from the above that $f_4(n)$ will grow faster than $f_2(n)$, such that $f_2(n) \leq O(f_4(n))$. This shows that $f_4(n)$ will be an upper bound of $f_2(n)$. Thus, $f_4(n) \geq \Omega(f_2(n))$, which is the lower bound.

Out of our set of functions we see that:

$$f_2(n),\ f_4(n),\ f_1(n),\ f_3(n)$$

The functions above are sorted from the slowest to the fastest growth rates.

$\square$

12.2 Problem 12(b)

(b) $(\log_7 n)^7$, $10\log_5 n$, $\log_7(n^7)$, $n^{1/1000}$. *Hint:* Recall change of logarithmic base formula $\log_a x = \log_b x \cdot \log_a b$

Answer. Let us first start by listing out the functions that we will be using:

$$f_1(n)=(\log_7 n)^7$$
$$f_2(n)=10\log_5 n$$
$$f_3(n)=\log_7(n^7)$$
$$f_4(n)=n^{1/1000}$$

I will first compare the functions of $f_1(n)$ and $f_3(n)$ using L'hopital's rule:

$$\begin{aligned}
\lim_{n\to\infty}\frac{(\log_7 n)^7}{\log_7(n^7)} &= \lim_{n\to\infty}\frac{\log_7 n)^7}{\log_7(n^7)} = \frac{\infty}{\infty} \qquad \mathit{Indeterminate}\\
&= \lim_{n\to\infty}\frac{\frac{7(\log_7 n)^6}{n\ln 7}}{\frac{7}{n\ln 7}}\\
&= \lim_{n\to\infty}\frac{7(\log_7 n)^6}{n\ln 7} * \frac{n\ln 7}{7}\\
&= \lim_{n\to\infty}\frac{7(\log_7 n)^6}{7} = \frac{\infty}{7}\\
&= \infty
\end{aligned}$$

We can see from the above that $f_1(n)$ will grow faster than $f_3(n)$, where $f_1(n)$ will be an upper bound of $f_3(n)$, such that $f_3(n) \le O(f_1(n))$. Thus, $f_1(n) \ge \Omega(f_3(n))$, which is the lower bound.

I will now compare f $f_1(n)$ and $f_2(n)$ using L'hopital's rule:
I will first change the base of $f_2(n)$ into $\log_7$ using the log base formula:

$$\begin{aligned}
10\log_5 n &= 10(\frac{\log_7 n}{\log_7 5})\\
&= (\frac{10}{\log_7 5})\log_7 n
\end{aligned}$$

I will now compare the two functions after performing the log base formula:

$$\begin{aligned}
\lim_{n\to\infty}\frac{(\log_7 n)^7}{10\log_5 n} &= \lim_{n\to\infty}\frac{\log_7 n)^7}{(\frac{10}{\log_7 5})\log_7 n} = \frac{\infty}{\infty} \qquad \mathit{Indeterminate}\\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{(\log_7 n)^7}{\log_7 n}\\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{(7(\log_7 n)^6) * \frac{1}{n\ln 7}}{\frac{1}{n\ln 7}}\\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}(7(\log_7 n)^6) = (\frac{10}{\log_7 5}) * \infty\\
&= \infty
\end{aligned}$$

We can see from the above that $f_1(n)$ will grow faster than $f_2(n)$, where $f_1(n)$ will be an upper bound of $f_2(n)$, such that $f_2(n) \le O(f_1(n))$. Thus, $f_1(n) \ge \Omega(f_2(n))$, which is the lower bound.

I will now compare the functions $f_1(n)$ with $f_4(n)$, using L'hopital's rule:

$$\lim_{n\to\infty} \frac{(\log_7 n)^7}{n^{\frac{1}{1000}}} = \lim_{n\to\infty} \frac{\log_7 n)^7}{n^{\frac{1}{1000}}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate}$$

I will now do the log base formula on $f_1(n)$:

$$(\log_7 n)^7 = (\frac{\ln n}{\ln 7})^7$$

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{(\frac{\ln n}{\ln 7})^7}{n^{\frac{1}{1000}}} \\
&= \frac{1}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^7}{n^{\frac{1}{1000}}} \\
&= \frac{1}{(\ln 7)^7} \lim_{n\to\infty} \frac{7(\ln n)^6 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{7000}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^6}{n^{\frac{1}{1000}}} \\
&= \frac{7000}{(\ln 7)^7} \lim_{n\to\infty} \frac{6(\ln n)^5 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{42,000,000}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^5}{n^{\frac{1}{1000}}} \\
&= \frac{42,000,000}{(\ln 7)^7} \lim_{n\to\infty} \frac{5(\ln n)^4 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{2.1 * 10^{11}}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^4}{n^{\frac{1}{1000}}} \\
&= \frac{2.1 * 10^{11}}{(\ln 7)^7} \lim_{n\to\infty} \frac{4(\ln n)^3 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{8.4 * 10^{14}}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^3}{n^{\frac{1}{1000}}} \\
&= \frac{8.4 * 10^{14}}{(\ln 7)^7} \lim_{n\to\infty} \frac{3(\ln n)^2 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{2.52 * 10^{18}}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^2}{n^{\frac{1}{1000}}} \\
&= \frac{2.52 * 10^{18}}{(\ln 7)^7} \lim_{n\to\infty} \frac{2(\ln n)^1 * \frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{5.04 * 10^{21}}{(\ln 7)^7} \lim_{n\to\infty} \frac{(\ln n)^1}{n^{\frac{1}{1000}}} \\
&= \frac{5.04 * 10^{21}}{(\ln 7)^7} \lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{1000} n^{\frac{-999}{1000}}} \\
&= \frac{5.04 * 10^{24}}{(\ln 7)^7} \lim_{n\to\infty} \frac{\frac{1}{n}}{n^{\frac{-999}{1000}}} = \frac{0}{\infty} = 0 \\
&= \frac{5.04 * 10^{24}}{(\ln 7)^7} * 0 \\
&= 0
\end{aligned}
$$

From the above we can see that the function $f_1(n)$ will grow slower than the function $f_4(n)$, such that the function $f_4(n)$ will be an upper bound of $f_1(n)$. This is because as $f_4(n)$ is approaching infinity it will become an upper bound of $f_1(n)$. Such that $f_1(n) \leq O(f_3(n))$. Since $f_4(n)$ grows faster than the function $f_1(n)$, we can infer that it will grow the fastest amongst our set of functions. Thus, $f_4(n) \geq \Omega(f_1(n))$, which is the lower bound.

I will now compare the functions of $f_2(n)$ and $f_3(n)$ to see out of these functions which one will grow faster using L'hopital's rule:

I will first change the base of $f_2(n)$ into $\log_7$ using the log base formula:

$$
\begin{aligned}
10\log_5 n &= 10(\frac{\log_7 n}{\log_7 5}) \\
&= (\frac{10}{\log_7 5})\log_7 n
\end{aligned}
$$

$$
\begin{aligned}
\lim_{n\to\infty}\frac{10\log_5 n}{\log_7(n^7)} &= \lim_{n\to\infty}\frac{(\frac{10}{\log_7 5})\log_7 n}{\log_7(n^7)} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{\log_7 n}{\log_7(n^7)} \\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{\frac{1}{n\ln 7}}{\frac{7}{n\ln 7}} \\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{1}{n\ln 7} * \frac{n\ln 7}{7} \\
&= (\frac{10}{\log_7 5}) * \lim_{n\to\infty}\frac{1}{7} \\
&= (\frac{10}{\log_7 5}) * \frac{1}{7} \\
&= (\frac{10}{7(\log_7 5)})
\end{aligned}
$$

We can see from the above that the functions $f_2(n)$ and $f_3(n)$ will grow at the same rate of each other since the limit is a constant. Such that it will be $f_2(n) \in \Theta(f_3(n))$.

Out of our set of functions we see that:

$$f_2(n),\ f_3(n),\ f_1(n),\ f_4(n)$$

The functions above are sorted from the slowest to the fastest growth rates.
The functions of $f_2(n)$ and $f_3(n)$ will grow at the same asymptotic rate such that $f_2(n) \in \Theta(f_3(n))$ □

13 Standard 13- Asymptotics II (Calculus II techniques):

Problem 13. *For each of the following questions, put the growth rates in order, from slowest-growing to fastest. That is, if your answer is $f_1(n), f_2(n), \ldots, f_k(n)$, then $f_i(n) \leq O(f_{i+1}(n))$ for all i. If two adjacent ones are asymptotically the same (that is, $f_i(n) = \Theta(f_{i+1}(n))$), you must specify this as well. Justify your answer (show your work). You may assume transitivity: if $f(n) \leq O(g(n))$ and $g(n) \leq O(h(n))$, then $f(n) \leq O(h(n))$, and similarly for little-oh, etc. The same instructions as for Problem 1 apply.*

13.1 Problem 13(a)

(a) $n^{\log_5 n}$, $\quad 4$, $\quad n^{\log_3 n}$, $\quad n^{\log_n (n^2)}$, $\quad n^{\log_n 5}$.

Answer. I will first define the functions that we will be using:

$$\begin{aligned} f_1(n) &= n^{\log_5 n} \\ f_2(n) &= 4 \\ f_3(n) &= n^{\log_3 n} \\ f_4(n) &= n^{\log_n (n^2)} \\ f_5(n) &= n^{\log_n 5} \end{aligned}$$

We will first compare functions $f_1(n)$ and $f_5(n)$ to see which function will grow faster:

$$\begin{aligned} &= \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^{\log_n 5}} = \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^{\log_n 5}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\ &= \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^{\log_n 5}} \\ &= \lim_{n\to\infty} \frac{n^{\log_5 n}}{5} \\ &= \lim_{n\to\infty} \frac{n^{\log_5 n}}{5} \\ &= \frac{\infty}{5} \\ &= \infty \end{aligned}$$

We can see from the above that the function $f_1(n)$ will grow faster than the function $f_5(n)$. Such that $f_5(n) \leq O(f_1(n))$. Further, the function $f_1(n)$ will be an upper bound of $f_5(n)$. Thus, the function $f_1(n) \geq \Omega(f_5(n))$, such that it will be a lower bound.

I will now compare the functions of $f_2(n)$ and $f_5(n)$, to see which function will grow faster:

$$\begin{aligned} &= \lim_{n\to\infty} \frac{4}{n^{\log_n 5}} \\ &= \lim_{n\to\infty} \frac{4}{5} \\ &= \frac{4}{5} \end{aligned}$$

We can see from the above that the functions of $f_2(n)$ and $f_5(n)$ will grow at the same rate as each other, such that they will be big theta of one another. Such that it will be $f_2(n) \in \Theta(f_5(n))$.

I will now compare the functions of $f_3(n)$ and $f_1(n)$ to see which function will grow faster:
I will change the base of function $f_3(n)$ using the log base change formula:

$$n^{\log_3 n} = \frac{\log_5 n}{\log_5 3}$$

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{n^{\log_3 n}}{n^{\log_5 n}} = \lim_{n\to\infty} \frac{n^{\log_3 n}}{n^{\log_5 n}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\
&= \lim_{n\to\infty} n^{\log_3 n - \log_5 n} \\
&= \lim_{n\to\infty} n^{\frac{1}{\log_5 3} - \log_5 n} \\
&= \lim_{n\to\infty} \frac{1}{\log_5 3} - 1 \geq 1 \\
&= \infty^{\infty} \\
&= \infty
\end{aligned}
$$

We can see from the above that the function $f_3(n)$ will grow faster than the function of f $f_1(n)$, such that it will become $f_1(n) \leq O(f_3(n))$, which will become an upper bound. Further, the function $f_3(n) \leq \Omega(f_1(n))$.

I will now compare the functions of $f_3(n)$ and $f_4(n)$ to see which function will grow faster:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{n^{\log_3 n}}{n^{\log_n (n^2)}} = \lim_{n\to\infty} \frac{n^{\log_3 n}}{n^{\log_n (n^2)}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\
&= \lim_{n\to\infty} \frac{n^{\log_3 n}}{n^2} \\
&= \lim_{n\to\infty} n^{\log_3 n - 2} \\
&= \lim_{n\to\infty} n^{\log_3 n - \log_3 9} \\
&= \lim_{n\to\infty} n^{\log_3 \frac{n}{9}} \\
&= \infty^{\infty} \\
&= \infty
\end{aligned}
$$

From the above we can see that the function $f_3(n)$ will grow faster than the function $f_4(n)$, such that $f_4(n) \leq O(f_3(n))$. Where the function $f_3(n)$ will be an upper bound of $f_4(n)$. Such that $f_3(n) \leq \Omega(f_4(n))$.

I will now compare the function of $f_1(n)$ and $f_4(n)$ to see which function will grow faster:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^{\log_n (n^2)}} = \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^{\log_n (n^2)}} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\
&= \lim_{n\to\infty} \frac{n^{\log_5 n}}{n^2} \\
&= \lim_{n\to\infty} n^{\log_5 n - 2} \\
&= \lim_{n\to\infty} n^{\log_5 n - \log_5 25} \\
&= \lim_{n\to\infty} n^{\log_5 \frac{n}{25}} \\
&= \infty^{\infty} \\
&= \infty
\end{aligned}
$$

We can see from the above that the function of $f_1(n)$ will grow faster than the function of $f_4(n)$, such that $f_4(n) \leq O(f_1(n))$, where the function $f_1(n)$ will be an upper bound of the function $f_4(n)$. Further, this will cause for the function $f_4(n)$, to be a lower bound of the function $f_1(n)$, such that $f_1(n) \leq \Omega(f_4(n))$.

Lastly, I will compare the function of $f_4(n)$ and $f_2(n)$, since we know that the function of $f_2(n)$ and $f_5(n)$ will grow at the same rate we can just compare for one of these functions:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{n^{\log_n(n^2)}}{4} = \lim_{n\to\infty} \frac{n^{\log_n(n^2)}}{4} = \frac{\infty}{\infty} \qquad \textit{Indeterminate} \\
&= \lim_{n\to\infty} \frac{n^{\log_n(n^2)}}{4} \\
&= \lim_{n\to\infty} \frac{n^2}{4} \\
&= \lim_{n\to\infty} \frac{\infty}{4} \\
&= \infty
\end{aligned}
$$

We can see from the above that the function $f_4(n)$ will grow faster than the function $f_2(n)$ and likewise $f_5(n)$ as well. Such that the function $f_2(n) \leq O(f_4 n))$ and $f_5(n) \leq O(f_4(n))$. Both the functions $f_2(n)$ and $f_5(n)$ will be the lower bound of the function $f_4(n)$. Thus, $f_4(n) \leq \Omega(f_2(n))$ and $f_4(n) \leq \Omega(f_5(n))$.

The final answer after performing the necessary calculation shall be :

$$f_5(n),\ f_2(n),\ f_4(n),\ f_1(n),\ f_3(n)$$

The functions above are sorted from the slowest to the fastest growth rates.
The functions of both $f_2(n)$ and $f_5(n)$ will grow at the same asymptomatic rate such that $f_2(n) = \Theta(f_5(n))$
□

13.2 Problem 13(b)

(b) $n!$, 2^n , $2^{n/3}$, n^n, 2^{n-2}, $\sqrt{n^{2n+1}}$. (*Hint:* Recall Stirling's approximation, which says that $n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$, i.e. $\lim_{n\to\infty} \frac{n!}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} = 1$.).

Answer. I will first define the functions that we will be using to compare:

$$f_1(n)\text{: } n! \ f_2(n)\text{: } 2^n \ f_3(n)\text{: } 2^{n/3} \ f_4(n)\text{: } n^n \ f_5(n)\text{: } 2^{n-2} \ f_6(n)\text{: } \sqrt{n^{2n+1}}$$

I will first compare the functions of $f_2(n)$ and $f_3(n)$ to see which function will grow faster using the ratio test:

$$\begin{aligned}
&= \lim_{n\to\infty} \frac{2^n}{2^{\frac{n}{3}}} \\
&= \lim_{n\to\infty} \frac{2^{n+1}}{2^{\frac{n+1}{3}}} * \frac{2^{\frac{n}{3}}}{2^n} \\
&= \lim_{n\to\infty} \frac{2^n * 2}{2^{\frac{n}{3}} * 2^{\frac{1}{3}}} * \frac{2^{\frac{n}{3}}}{2^n} \\
&= \lim_{n\to\infty} \frac{2}{2^{\frac{1}{3}}} \\
&= \lim_{n\to\infty} \frac{2}{\sqrt[3]{2}} \\
&= \frac{2}{\sqrt[3]{2}} > 1
\end{aligned}$$

From this we can see that this limit will be divergent, by the ratio test such that the function $f_2(n)$ will grow faster than the function of $f_3(n)$, such that $f_3(n) \leq O(f_2(n))$. Where $f_3(n)$ exists as a lower bound of $f_2(n)$, such that $f_3(n) \leq \Omega(f_2(n))$.

I will now compare the functions of $f_2(n)$ and $f_1(n)$, to see which function will grow faster using the ratio test:

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

From this we can see that this limit will be convergent, by the ratio test such that the function $f_1(n)$ will grow faster than the function $f_2(n)$. Such that $f_2(n) \leq O(f_1(n))$. Where $f_2(n)$ will exist as a lower bound to the function $f_1(n)$, such that $f_1(n) \leq \Omega(f_2(n))$.

I will now compare the functions of $f_4(n)$ and $f_2(n)$, to see which function will grow faster by the ratio test:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{2^n}{n^n} \\
&= \lim_{n\to\infty} \frac{2^{n+1}}{(n+1)^{n+1}} * \frac{n^n}{2^n} \\
&= \lim_{n\to\infty} \frac{2^n * 2}{n+1^{n+1}} * \frac{n^n}{2^n} \\
&= \lim_{n\to\infty} \frac{2}{n+1^{n+1}} * n^n \\
&= \lim_{n\to\infty} \frac{2}{n+1} * (\frac{n}{n+1})^n \\
&= \lim_{n\to\infty} \frac{2}{n+1} * (\frac{1^n}{1+\frac{1}{n}^n}) \qquad (1+\frac{1}{n})^n = e \\
&= \lim_{n\to\infty} \frac{2}{n+1} * (\frac{1}{e}) \\
&= \lim_{n\to\infty} \frac{2}{en+e} \\
&= \frac{2}{\infty} \\
&= 0 < 1
\end{aligned}
$$

From this we can see that this limit will be convergent, by the ratio test such that the function $f_4(n)$ will grow faster than the function $f_2(n)$. Such that $f_2(n) \leq O(f_4(n))$. Where $f_2(n)$ will exist as a lower bound to the function $f_2(n)$, such that $f_4(n) \leq \Omega(f_2(n))$.

I will now compare the functions $f_4(n)$ and $f_6(n)$, using the ratio test:

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

From this we can see that this limit will be convergent, by the ratio test such that the function $f_6(n)$ will grow faster than the function $f_4(n)$. Such that $f_4(n) \leq O(f_6(n))$. Where $f_4(n)$ will exist as a lower bound to the function $f_6(n)$, such that $f_6(n) \leq \Omega(f_4(n))$.

I will now compare the functions $f_1(n)$ and $f_4(n)$, using L'hopital's rule to determine it as well as Sterling's approximation since we know that $n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{n!}{n^n} \\
&= \lim_{n\to\infty} \frac{(\frac{n}{e})^n * \sqrt{2\pi n}}{n^n} \\
&= \lim_{n\to\infty} \frac{(\frac{1}{e}^n) * (n^n) * (\sqrt{2\pi}) * (\sqrt{n})}{n^n} \\
&= (\sqrt{2\pi}) * \lim_{n\to\infty} (\frac{1}{e}^n) * (n^{\frac{1}{2}}) \\
&= (\sqrt{2\pi}) * \lim_{n\to\infty} (\frac{n^{\frac{1}{2}}}{e^n}) \\
&= (\sqrt{2\pi}) * \lim_{n\to\infty} (\frac{\frac{1}{2}n^{-\frac{1}{2}}}{e^n}) \\
&= (\sqrt{2\pi}) * \lim_{n\to\infty} (\frac{1}{2e^{n^{\frac{1}{2}}}}) \\
&= (\sqrt{2\pi}) * 0 \\
&= 0
\end{aligned}
$$

From the above we know that $f_4(n)$ will grow faster than $f_1(n)$, by L'hopital's rule and using Sterling's approximation, we know that the function $f_4(n)$ will be an upper bound of $f_1(n)$, such that $f_1(n) \leq O(f_4(n))$. We know that the function $f_1(n)$ will be a lower bound of $f_4(n)$, such that $f_4(n) \leq \Omega(f_1(n))$.

I will now compare $f_2(n)$ and $f_5(n)$ to see which function will grow faster by using the ratio test:

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{2^n}{2^{n-2}} \\
&= \lim_{n\to\infty} \frac{2^{n+1}}{2^{n-1}} * \frac{2^{n-2}}{2^n} \\
&= \lim_{n\to\infty} \frac{2 * 2^n}{2^n * 2^{-1}} * \frac{2^n * 2^{-2}}{2^n} \\
&= \lim_{n\to\infty} \frac{2}{\frac{1}{2}} * \frac{1}{4} \\
&= \lim_{n\to\infty} \frac{2}{4 * \frac{1}{2}} \\
&= \lim_{n\to\infty} \frac{2}{2} \\
&= 1 = 1
\end{aligned}
$$

From the above we see that the limit of the functions $f_2(n)$ and $f_5(n)$, by the ratio test will be inconclusive so we will have to use a different test, I will find a constant between the two functions to see which function will grow faster:

$$f_5(n) = 2^{n-2}$$

$$
\begin{aligned}
2^{2n-2} &= (2^n) * (2^{-2}) \\
&= \frac{1}{4} * (2^n)
\end{aligned}
$$

The above shows that a constant exists between the functions of $f_2(n)$ and $f_5(n)$, such that:

$$\frac{1}{4} * f_2(n)$$
$$\frac{1}{4} * 2^n)$$
$$f_2(n)=\Theta(f_5(n))$$

From the above we can see that a constant exists between the functions of $f_2(n)$ and $f_5(n)$, since there exists a constant this means that these two functions will grow at the same rate as one another. From this it is shown that $f_2(n)=\Theta(f_5(n))$.

Since we know that the function $f_2(n)$ will grow faster than the function $f_3(n)$, since both the function of $f_2(n)$ and $f_5(n)$ grow at the same rate than the function $f_5(n)$ must also grow faster than the function $f_3(n)$.

The final answer after performing the necessary calculations shall be :

$$f_3(n),\ f_5(n),\ f_2(n),\ f_1(n),\ f_4(n),\ f_6(n)$$

The functions above are sorted from the slowest to the fastest growth rates.
The functions of both $f_2(n)$ and $f_5(n)$ will grow at the same asymptomatic rate such that $f_2(n) = \Theta(f_5(n))$
$\square$

14 Standard 14- Analyzing Code I: (Independent nested loops)

Problem 14. Analyze the *worst-case* runtime of the following algorithms. Clearly derive the runtime complexity function $T(n)$ for this algorithm, and then find a tight asymptotic bound for $T(n)$ (that is, find a function $f(n)$ such that $T(n) = \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

Algorithm 1 Nested Algorithm 1

```
1: procedure Foo2(Integer n)
2:     for i ← 1; i ≤ n; i ← i + 2 do
3:         for j ← 1; j ≤ n; j ← 3 * j do
4:             print "Hi"
```

Answer. **Referenced Levet notes.**
We will first begin by analyzing the j loop that is present on line 3, such that:

The j-loop will take one step to initialize j=1.

Further from this line I will find how long the loop will take to run:

We can see that the loop will be for each iteration:
$n \leq 3^k$ We can solve for k in this case by taking the log of both sides:
$k = \log_3 n$

From here we can see that the comparison of j and n will cause for at least one iteration to be added such that:

$$k = \log_3 n + 1$$

I will now find what the loop will be doing at each iteration:
-The comparison of $j \leq n$ will add 1 step
- On line 3 the update $j \leftarrow 3 * j$ will take 2 steps, this is because the 1 step will be used to evaluate $3 * j$ and one more step for the assignment of $j \leftarrow$
- Within this loop it contains the print statement of "Hi" which will create 1 more step.

From this we can find the inner loop's time complexity:

$$\begin{aligned}
1 + \sum_{j=1}^{\log_3 n+1} (1+2+1) &= 1 + \sum_{j=1}^{\log_3 n+1} 4 \\
&= 1 + 4(\log_3 n + 1) \\
&= (4\log_3 n + 4) + 1 \\
&= 4\log_3 n + 5
\end{aligned}$$

From the above we can see that this will be the runtime-complexity of the inner loop.
I will now analyze the outer loop to find the runtime complexity of that loop, such that:

From this loop we can see that the i-loop will take 1 step to initialize $i = 1$.

Further from this I will find how long it will take for this loop to iterate:

We can see from this loop that the terminating term will be n such that:
$1 + 2k > n$, solving for k we get $k = \frac{n-1}{2}$

Since the comparison of $i \leq n$ will cause for one step to be added such that:

$$k = \frac{n-1}{2} + 1$$

□

I will now find what the loop will be doing at each iteration:
-The comparison of $i \leq n$ will add 1 step
- On line 3 the update $i \leftarrow i + 2$ will take 2 steps, this is because the 1 step will be used to evaluate $i + 2$ and one more step for the assignment of $i \leftarrow$
- The main body of the i-loop will be contained within the j-loop

I will now find the time complexity of T(n) with both the loop of i and j:

$$\begin{aligned}
T(n) &= 1 + \sum_{i=1}^{\frac{n-1}{2}} (1 + 2) \\
&= 1 + \sum_{i=1}^{\frac{n-1}{2}} 3 + 4\log_3 n + 5 \\
&= 1 + \sum_{i=1}^{\frac{n-1}{2}} 8 + 4\log_3 n \\
&= 1 + \sum_{i=1}^{\frac{n-1}{2}} 8 + \sum_{i=1}^{\frac{n-1}{2}} 4\log_3 n \\
&= 1 + (\frac{n-1}{2}) * 8 + (\frac{n-1}{2}) * (4\log_3 n) \\
&= 1 + (\frac{8n-8}{2}) + (\frac{4\log_3 n * n - 1}{2}) \\
&= 1 + (4n - 4) + (\frac{4n\log_3 n - 4\log_3 n}{2}) \\
&= 1 + (4n - 4) + 2n\log_3 n - 2\log_3 n
\end{aligned}$$

We can see from this that the runtime complexity for T(n):

$$T(n) = \Theta(n \log_3 n)$$

15 Standard 15- Analyzing Code II: (Dependent nested loops)

Problem 15. Analyze the *worst-case* runtime of the following algorithms. Clearly derive the runtime complexity function $T(n)$ for this algorithm, and then find a tight asymptotic bound for $T(n)$ (that is, find a function $f(n)$ such that $T(n) = \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

Algorithm 2 Nested Algorithm 2

```
1: procedure Boo1(Integer n)
2:     for i ← 1; i ≤ n; i ← i + 1 do
3:         for j ← i; j ≤ n; j ← j + 1 do
4:             print "Hi"
```

Answer. **Referenced Levet Notes**
For dependently nested loops the inner loop will be based on what the outside loop does. We will first begin by analyzing the j loop that is present on line 3, such that:

The j-loop will take one step to initialize j=1.

I will now find what the loop will be doing at each iteration:
-The comparison of $j \leq n$ will add 1 step
- On line 3 the update $j \leftarrow j + 1$ will take 2 steps, this is because the 1 step will be used to evaluate $j + 1$ and one more step for the assignment of $j \leftarrow$
- Within this loop it contains the print statement of "Hi" which will create 1 more step.

From this we can find the amount of steps total that the inner loop will take to run:

$$\begin{aligned} 1 + \sum_{j=i}^{n} (1 + 2 + 1) &= 1 + \sum_{j=i}^{n} 4 \\ &= 1 + 4(n - i) \end{aligned}$$

I will now analyze the outer i loop to find what the runtime will be for that loop:

From this loop on line 2 we can see that the i-loop will take 1 step to initialize $i = 1$.

I will now find what the loop will be doing at each iteration:
-The comparison of $i \leq n$ will add 1 step
- On line 3 the update $i \leftarrow i + 2$ will take 2 steps, this is because the 1 step will be used to evaluate $i + 1$ and one more step for the assignment of $i \leftarrow$
- The main body of the i-loop will be contained within the j-loop

I will now find the time complexity of T(n) with both the loop of i and j:

$$
\begin{aligned}
1 + \sum_{i=1}^{n}(1+2) &= 1 + \sum_{i=1}^{n}(1+2) + (1 + 4(n-i)) \\
&= 1 + \sum_{i=1}^{n} 4 + 4(n-i) \\
&= 1 + \sum_{i=1}^{n} 4 + 4\sum_{i=1}^{n} n - i \\
&= 1 + \sum_{i=1}^{n} 4 + 4\sum_{i=1}^{n} n - 4\sum_{i=1}^{n} i \\
&= 1 + 4n + 4n^2 - 4(\frac{n(n+1)}{2})
\end{aligned}
$$

From this we can see that the runtime complexity for T(n):

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

$\square$