CSCI 3104 Spring 2022
Instructor: Profs. Chen and Layer

Problem Set 4

Due Date .... February 15
Name .... **Your Name**
Student ID .... **Your Student ID**
Collaborators .... **List Your Collaborators Here**

Contents

1 Instructions

- The solutions **must be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- 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 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.
- 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.

Agreed (signature here). I agree to the above. Ethan Richman □

3 Standard 10- Asymptotics I (Calculus I techniques)

Problem 2. 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 $f_i(n) \in O(f_{i+1}(n))$ for all i. If two adjacent functions have the same order of growth (that is, $f_i(n) \in \Theta(f_{i+1}(n))$), 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 $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$. The same applies for 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.

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

3.1 Problem 2(a)

(a) $n^3 - 10, \qquad n^3 + 20n^2 + 1000, \qquad n^4 - 50n^3, \qquad 10n^3\sqrt{n}.$

Answer.

- functions $n^3 + 2n^2 + 1000$ and $n^3 - 10$ grow at the same asymptotic rate but grow the slowest of all functions
- $10n^3\sqrt{n}$ grows faster than $n^3 + 2n^2 + 1000$ and $n^3 - 10$ but slower than $n^4 - 50n^3$
- $n^4 - 50n^3$ grows at the fastest rate

Limit Comparison Test Work:

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^4 - 50n^3}{10n^3\sqrt{n}} &= \frac{1}{10} \lim_{n\to\infty} \frac{n^4 - 50n^3}{n^3\sqrt{n}} \\
&= \frac{1}{10} \lim_{n\to\infty} \frac{n - 50}{\sqrt{n}} \\
&= \frac{1}{10} \lim_{n\to\infty} \sqrt{n} - \frac{50}{\sqrt{n}} \\
&= \frac{1}{10} \lim_{n\to\infty} \sqrt{n} - \lim_{n\to\infty} \frac{50}{\sqrt{n}} \\
&= \frac{1}{10}(\infty - 0) \\
&= \infty
\end{aligned}$$

$$\begin{aligned}
\lim_{n\to\infty} \frac{10n^3\sqrt{n}}{n^3 - 10} &= 10 \lim_{n\to\infty} \frac{n^3\sqrt{n}}{n^3 - 10} \\
&= 10 \lim_{n\to\infty} \frac{\sqrt{n}}{1 - \frac{10}{n^3}} \\
&= 10 \frac{\lim_{n\to\infty} \sqrt{n}}{\lim_{n\to\infty} 1 - \frac{10}{n^3}} \\
&= 10(\frac{\infty}{1}) \\
&= \infty
\end{aligned}$$

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^3 - 10}{n^3 + 2n^2 + 1000} &= \lim_{n\to\infty} \frac{1 - \frac{10}{n^3}}{1 + \frac{2}{n} + \frac{1000}{n^3}} \\
&= \frac{\lim_{n\to\infty} 1 - \frac{10}{n^3}}{\lim_{n\to\infty} 1 + \frac{2}{n} + \frac{1000}{n^3}} \\
&= \frac{1}{1} \\
&= 1
\end{aligned}$$

$\square$

3.2 Problem 2(b)

(b) $10 \log_2 n^3$, $(\log_3 n)^3$, $100 \log_4 n$, $n^{1/1000}$.

Hint: Recall change of logarithmic base formula $\log_a x = \log_b x \cdot \log_a b$

Answer.

- $10 \log_2 n^3$ is the slowest
- $100 \log_4 n$ is the next slowest
- $(\log_3 n)^3$ is the second fastest
- $n^{1/1000}$ is the fastest

Using L'Hopitals rule

$$\begin{aligned}
\lim_{n\to\infty} \frac{10 \log_2 n^3}{(\log_3 n)^3} &= \lim_{n\to\infty} \frac{30 \log_2 n}{3 \log_3 n} \\
&= \lim_{n\to\infty} \frac{10 \log_2 n}{log_3 n} \\
&= \lim_{n\to\infty} \frac{\frac{d}{dn}(10 \log_2 n)}{\frac{d}{dn} \log_3 n} \\
&= \lim_{n\to\infty} \frac{nln3}{nln2} \\
&= lim_{n\to\infty} \frac{\frac{d}{dn}(ln3)}{\frac{d}{dn}(ln2)} \\
&= 0
\end{aligned}$$

Therefore, $\log_3 n^3$ is greater than $10 \log_2 n^3$.

$$\begin{aligned}
\lim_{n\to\infty} \frac{100 \log_4 n}{(\log_3 n)^3} &= \frac{100 \log_2 n}{3 \log_3 n} \\
&= \lim_{n\to\infty} \frac{\frac{d}{dn}(100 \log_2 n)}{\frac{d}{dn}(3 \log_3 n)} \\
&= \lim_{n\to\infty} \frac{100 n \ln 3}{3 n \ln 2} \\
&= \lim_{n\to\infty} \frac{\frac{d}{dn} 100 \ln 3}{\frac{d}{dn} 3 \ln 2} \\
&= 0
\end{aligned}$$

Therefore, $\log_3 n^3$ is greater than $100 \log_4 n$.

$$\begin{aligned}
\lim_{n\to\infty} \frac{10 \log_2 n}{100 \log_4 n} &= \lim_{n\to\infty} \frac{30 \log_2 n}{100 \log_4 n} \\
&= \lim_{n\to\infty} \frac{\frac{d}{dn}(30 \log_2 n)}{\frac{d}{dn}(100 \log_4 n)} \\
&= \lim_{n\to\infty} \frac{30}{n \ln 2} * \frac{n \ln 4}{100} \\
&= \lim_{n\to\infty} \frac{\frac{d}{dn}(30 \ln 4)}{\frac{d}{dn}(100 \ln 2)} \\
&= 0
\end{aligned}$$

Therefore, $100 \log_4 n$ is greater than $10 log_2 n^3$.

$$
\begin{aligned}
\lim_{n\to\infty} \frac{(\log_3 n)^3}{n^{1/1000}} &= \lim_{n\to\infty} \frac{(\log_3 3)^3}{n^{1/1000}} \\
&= \lim_{n\to\infty} \frac{n^{-1/1000}(\log_3 n)^3}{(log_3 3)^3} \\
&= \frac{1}{(\log_3 3)^3} \lim_{n\to\infty} n^{-1/1000}(\log_3 n)^3 \\
&= \frac{0}{(\log_3 3)^3} \\
&= 0
\end{aligned}
$$

Therefore, $n^{1/1000}$ is greater than $(\log_3 n)^3$. $\square$

4 Standard 11- Asymptotics II (Calculus II techniques):

Problem 3. *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* $f_i(n) \in O(f_{i+1}(n))$ *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 use transitivity: if* $f(n) \in O(g(n))$ *and* $g(n) \in O(h(n))$*, then* $f(n) \in O(h(n))$*, and similarly for Big-Theta, etc. The same instructions as for Problem 1 apply.*

4.1 Problem 3(a)

(a) 1 , $n^{\log_5 n^2}$, $n^{\log_2 n}$, $n^{\log_n (n^3)}$, $n^{\log_n 10}$.

Answer.

- functions 1 and $n^{\log_n 10}$ grow at the same rate and the slowest
- $n^{\log_n (n^3)}$ grows the next slowest
- $n^{\log_5 n^2}$ is the next fastest
- $n^{\log_2 n}$ is the fastest

□

4.2 Problem 3(b)

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

Answer.

- The function $3^{n/2}$ is the slowest
- The functions 3^{n-2} and 3^n grow at the same rate but slower than n!, n^n and $\sqrt{n^{3n+1}}$
- n! is the next fastest
- Followed by n^n
- The fastest function is $\sqrt{n^{3n+1}}$

Justification:

$$\begin{aligned}\lim_{n\to\infty} \frac{3^{n/2}}{3^n} &= \lim_{n\to\infty} 3^{\frac{n}{2}-n} \\ &= \lim_{n\to\infty} 3^{\frac{-n}{2}} \\ &= \lim_{n\to\infty} \frac{1}{3^{n/2}} \\ &= 0\end{aligned}$$

Therefore 3^n grows faster than $3^{n/2}$

$$\begin{aligned}\lim_{n\to\infty} \frac{3^n}{3^{n-2}} &= \lim_{n\to\infty} 3^{n-(n-2)} \\ &= \lim_{n\to\infty} 3^2 \\ &= 9\end{aligned}$$

Therefore 3^n and 3^{n-2} grow at the same rate.

$$\begin{aligned}\lim_{n\to\infty} \frac{3^n}{n!} &= \lim_{n\to\infty} \frac{3^{n+1} * n!}{(n+1)! * 3^n} \\ &= \lim_{n\to\infty} \frac{3}{n+1} \\ &= 0\end{aligned}$$

Therefore n^n grows faster than n!

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

Therefore n^n grows faster than n!

$$\begin{aligned}\lim_{n\to\infty}\frac{n^n}{\sqrt{n^{3n+1}}} &= \lim_{n\to\infty}\frac{n^n}{n^{\frac{3n+1}{2}}}\\ &= \lim_{n\to\infty} n^{n-\frac{3n+1}{2}}\\ &= \lim_{n\to\infty} n^{\frac{-n+1}{2}}\\ &= \lim_{n\to\infty}\frac{1}{n^{\frac{n+1}{2}}}\\ &= 0\end{aligned}$$

Therefore $\sqrt{n^{3n+1}}$ grows faster than n^n

$\square$

5 Standard 12- Analyzing Code I: (Independent nested loops)

Problem 4. 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 $T(n) \in \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

```
1: procedure foo_i(integer n):
2:   for i = 1, i <= n
3:       i = 3 * i
4:       print 'outer'
5:       for j = 1, j <= n
6:           j = 2 * j
7:           print 'inner'
```

Answer. At line 5 the inner printing j loop takes 1 step to initialize
This loop terminates when $2^k > n$. For solving for k we obtain that k¿$\log_2 n$. As the loop takes at least one iteration to compare j to n, we have to have the loop taking $(k > \log_2 n)$+1 iterations. In each iteration the following happens

- In each iteration the comparison j$\leq$ n takes one step.
- The update of j=2*j takes two steps, one for the math and one for the new assignment of j
- The body of the loop is a single print statement which takes 1 step

So the run time complexity of the inner j-loop is $1 + \sum_{j=1}^{(\log_2 n)+1} 1 + 2 + 1 = \sum_{j=1}^{(\log_2 n)+1} 4 = 1 + 4((\log_2 n) + 1) = 5 + 4(\log_2 n)$

Now we need to analyze the outer i loop.

- Initializing this loop takes 1 step
- This loop runs for $(log_3 n)$+1 iterations, in each iteration there is:
- The comparison of i$\leq$ n which is 1 step
- The 2 steps to evaluate i=3*i and the assignment of i to this new value
- The 1 step to print outer
- and the j loop

So we can analyze the code as a whole with the equation: $T(n) = 1 + \sum_{i=1}^{(\log_3 n} 1 + 2 + 1 + 5 + 4(log_2 n)$
$T(n) = 1 + \sum_{i=1}^{(\log_3 n} 9 + 4(log_2 n)$
$T(n) = 1 + \sum_{i=1}^{(\log_3 n} 9 + \sum_{i=1}^{(\log_3 n} 4(log_2 n)$
$T(n) = 1 + 9(log_3 n) + 4(log_2 n)(log_3 n)$
So $T(n) \in \Theta(n^2)$ □

6 Standard 13- Analyzing Code II: (Dependent nested loops)

Problem 5. 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 $T(n) \in \Theta(f(n))$). Avoid heuristic arguments from 2270/2824 such as multiplying the complexities of nested loops.

```
procedure foo_d(integer n):
  for i = 1, i <= n
      i = i + 1
      print ‘outer’
      for j = 1, j <= i
          j = j + 2
          print ‘inner’
```

Answer. We are starting by analyzing the inner loop. For this analysis we are imagining that i is a fixed number at this point. The j loop does the following:
The initialization of the loop takes 1 step.
$1 + 2k > n = 2k > n - 1 = k > (n-1)/2$
At each iteration the loop does the following:

- The comparison j≤i takes one step
- The update j = j+2 takes 2 steps, one for the evaluation of the equation and one for the setting of j's value
- The print statement takes 1 step

So the runtime complexity of the j-loop is: $\sum_{j=1}^{((i-1)/2)+1}(1+2+1) = \sum_{j=1}^{((i-1)/2)+1} 4 = 1 + (4((i-1)/2)+1) = 5 + (4((i-1)/2))$
We now turn to analyzing the outer i-loop. Initializing the loop takes 1 step. Observe that the i loop takes n iterations. At each iteration the loop does the following

- The comparison $i \leq n$ takes 1 step
- The update of i to i+1 takes 2 steps, one for the math and one for the reassignment
- The printing of the word outer takes 1 step
- the j loop is run inside of the i loop

So the run time complexity of $T(n)$ is: $T(n) = 1 + \sum_{i=1}^{n}(1+2+1+(5+(4(i-1)/2)))$
$T(n) = 1 + \sum_{i=1}^{n}(9 + 4((i-1)/2)))$
$T(n) = 1 + \sum_{i=1}^{n} 9 + \sum_{i=1}^{n}((i-1)/2)$
$T(n) = 1 + 9n + 4n * ((n-1)/2)$
Thus $T(n) \in \Theta(n^2)$ $\square$