CSCI 3104 Fall 2021
Profs Josh Grochow and Bo Waggoner

Problem Set 5

Due Date .... TODO
Name .... **John Blackburn**
Student ID **Jobl2177**
Collaborators .... **NA**

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, John Blackburn, have upheld the honor code □

1 Standard 12- Asymptotics I (Calculus I techniques)

Problem 1. 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 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.

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

1.1 Problem 1(a)

(a) $n^2 + 20n, \qquad n^2 - 100, \qquad n^3 - 10n^2, \qquad n^2\sqrt{n}.$

Answer. I will find the limit of two functions to determine the run time relationship of n^3-10n^2 and n^2+20n.

$$\lim_{n\to\infty} \frac{n^3 - 10n^2}{n^2 + 20n}$$

The limit of this is equivalent to $\frac{\infty}{\infty}$ therefore we can use L'hopitals rule to simplify the limit. We take the derivative of the top and bottom of the limit and get:

$$\lim_{n\to\infty} \frac{3n^2 - 20n}{2n + 20}$$

We still get $\frac{\infty}{\infty}$ so we do L'hopitals rule again. Take the derivative of both the top and bottom again resulting in:

$$\lim_{n\to\infty} \frac{6n - 20}{2} = \frac{\infty}{2} = \infty$$

We can now take the limit and get a proper result. We get that the limit is equivalent to ∞. Therefore because we used L'hopitals rule we can see that the original limit is also equal to ∞.

$$\lim_{n\to\infty} \frac{n^3 - 10n^2}{n^2 + 20n} = \infty$$

We know that when the limit results in ∞ as your result it means that we can determine that $n^3 - 10n^2 \in \Omega(n^2 + 20n)$. Therefore, we can determine that $n^3 - 10n^2$ grows faster than $n^2 + 20n$

The next comparison I will do is, I will solve the limit to compare $n^3 - 10n^2$ and $n^2\sqrt{n}$.

$$\lim_{n\to\infty} \frac{n^3 - 10n^2}{n^2\sqrt{n}} = \lim_{n\to\infty} \frac{n^3 - 10n^2}{n^{5/2}}$$

The first thing we can do is simplify the bottom of the fraction from $n^2\sqrt{n}$ to $n^{5/2}$. The next thing we can do is split up the fraction into two fractions like so:

$$\lim_{n\to\infty} \frac{n^3 - 10n^2}{n^{5/2}} = \lim_{n\to\infty} \frac{n^3}{n^{5/2}} - \frac{10n^2}{n^{5/2}}$$

From there we can simplify both fractions like so:

$$\lim_{n\to\infty} \frac{n^3}{n^{5/2}} - \frac{10n^2}{n^{5/2}} = \lim_{n\to\infty} n^{1/2} - \frac{10}{n^{1/2}}$$

From here we can finally take the limit by plugging in ∞ for all n in the function:

$$\lim_{n\to\infty} n^{1/2} - \frac{10}{n^{1/2}} = \infty - \frac{10}{\infty} = \infty - 0 = \infty$$

So the limit computes to ∞ meaning that $n^3 - 10n^2 \in \Omega(n^2\sqrt{n})$. Therefore we can determine that $n^3 - 10n^2$ grows faster than $n^2\sqrt{n}$.

So overall, we currently know that $n^3 - 10n^2 \in \Omega(n^2\sqrt{n})$ and $n^3 - 10n^2 \in \Omega(n^2 + 20n)$. From this we know that $n^3 - 10n^2$ is faster than both $n^2\sqrt{n}$ and $n^2 + 20n$. But we don't know the relationship between $n^2\sqrt{n}$ and $n^2 + 20n$. So next I will find the limit of $n^2\sqrt{n}$ and $n^2 + 20n$. We've already shown that $n^2\sqrt{n}$ can be simplified to $n^{5/2}$, so I will skip that step in my limit comparison test and start with:

$$\lim_{n\to\infty} \frac{n^2+20n}{n^{5/2}} = \lim_{n\to\infty} \frac{n^2}{n^{5/2}} - \frac{20n}{n^{5/2}}$$

I have broken up the fraction into individual components for easier computation of the limit and now I will further simplify the fractions and then plug in infinity to determine the limit.

$$\lim_{n\to\infty} \frac{n^2}{n^{5/2}} - \frac{20n}{n^{5/2}} = \lim_{n\to\infty} \frac{1}{n^{1/2}} + \frac{20}{n^{3/2}} = \frac{1}{\infty} + \frac{20}{\infty} = 0$$

After plugging in infinity we see that the limit comes to zero. This means that $n^2 + 20n \in O(n^2\sqrt{n})$. Therefore we know that $n^2\sqrt{n}$ grows faster than $n^2 + 20n$ and doesn't grow as fast as $n^3 - 10n^2$.

Finally, I will compare $n^2 + 20n$ and $n^2 - 100$ by fidning the limit of the first function over the second.

$$\lim_{n\to\infty} \frac{n^2+20n}{n^2-100}$$

I will use L'hopitals rule twice to simplify the limit

$$\lim_{n\to\infty} \frac{2n+20}{2n}$$

L'hop

$$\lim_{n\to\infty} \frac{2}{2} = 1$$

After using L'hop we get the limit is equal to 1. This means that $n^2 + 20n \in \theta(n^2 - 100)$. Therefore we can know that $n^2 + 20n$ runs at the same rate as $(n^2 - 100)$. Since we know $n^2 + 20n \in O(n^2\sqrt{n})$, and $n^2\sqrt{n} \in O(n^3 - 10n^2)$, we can determine the entire relationship between the functions through transitivity.

The slowest growing function is $n^2 - 100$, next is $n^2 + 20n$ however $n^2 - 100 \in \Theta(n^2 + 20n)$, so they run at the same asymptomatic rate. The next fastest growing function is $n^2\sqrt{n}$, and finally the fastest growing function is $n^3 - 10n^2$.

$\square$

1.2 Problem 1(b)

(b) $(\log_4 n)^2, \qquad 10\log_3 n, \qquad \log_2 n^2, \qquad n^{1/1000}.$

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

Answer. First thing I will do is compare $(\log_4 n)^2$ and $10\log_3 n$ by finding the limit of the first funciton over the second. Our starting point is:

$$\lim_{n\to\infty} \frac{(\log_4 n)^2}{10\log_3 n}$$

I will use L'hop to simplify the limit such that we can find the limit. After taking the derivative of the top and bottom the result is:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{10}{\ln 3(n)}}$$

after simplifying the above limit we reach:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{10}{\ln 3(n)}} = \lim_{n\to\infty} \frac{(\log_4 n)\ln 3}{10\ln 2}$$

Now we have a constant on the bottom of our equation so we can plug in infinity into n and that gives me:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{10}{\ln 3(n)}} = \lim_{n\to\infty} \frac{(\log_4 n)\ln 3}{10\ln 2} = \frac{\infty}{10\ln 2} = \infty$$

Our limit comes out to be infinity. this means that $(\log_4 n)^2 \in \Omega(10\log_3 n)$. Therefore we can determine that $(\log_4 n)^2$ runs faster than $10\log_3 n$.

For our next comparison I will find the limit of $10\log_3 n$ over $\log_2 n^2$. Our starting point is:

$$\lim_{n\to\infty} \frac{10\log_3 n}{\log_2 n^2}$$

I will use L'hop to simplify the limit. I take the derivative of the top and bottom and the result is:

$$\lim_{n\to\infty} \frac{\frac{(10)}{\ln 3(n)}}{\frac{2}{\ln 2(n)}}$$

Now I simplify the limit and the result is:

$$\lim_{n\to\infty} \frac{\frac{(10)}{\ln 3(n)}}{\frac{2}{\ln 2(n)}} = \lim_{n\to\infty} \frac{10\ln 2}{2\ln 3}$$

And since all the n variables cancelled out we know the limit is equivalent to the above constant. Therefore we know $10\log_3 n \in \theta(\log_2 n^2)$, and that $10\log_3 n$ and $\log_2 n^2$ run at the same rate. Through transitivity we can also determine that $(\log_4 n)^2 \in \Omega(\log_2 n^2)$.

For my next comparison i will compare $(\log_4 n)^2$ and $n^{1/1000}$. I will find the limit of the first function over the second and use L'hop to determine their relationship in terms on run time. My starting point is:

$$\lim_{n\to\infty} \frac{(\log_4 n)^2}{n^{1/1000}}$$

Now I use L'hop to take the derivative of the top and bottom. result is:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{1}{1000n^{999/1000}}}$$

Now simplifying results in:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{1}{1000n^{999/1000}}} = \lim_{n\to\infty} \frac{1000(\log_4 n)}{\ln 2(n^{1/1000})}$$

Now we can plug in infinity. This gives us:

$$\lim_{n\to\infty} \frac{\frac{(\log_4 n)}{\ln 2(n)}}{\frac{1}{1000n^{999/1000}}} = \lim_{n\to\infty} \frac{1000(\log_4 n)}{\ln 2(n^{1/1000})} = \frac{1000(\log_4 n)}{\infty} = 0$$

Our limit is zero which means $(\log_4 n)^2 \in O(n^{1/1000})$. which means that $(\log_4 n)^2$ runs slower than $n^{1/1000}$. Through transitivity, since we know $(\log_4 n)^2$ runs faster than all the other functions not including $n^{1/1000}$, we know $n^{1/1000}$ is faster than all the functions in this group.

We know now enough to determine that $10\log_3 n$, and $\log_2 n^2$ are the slowest functions in the group, and that they run at the same rate as concluded above when we discovered that $10\log_3 n \in \theta(\log_2 n^2)$. The next fastest is $(\log_4 n)^2$, then finally the fastest is $n^{1/1000}$.

$\square$

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

Problem 2. *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.*

2.1 Problem 2(a)

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

Answer. $\square$

2.2 Problem 2(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. □

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

Problem 3. 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.

Algorithm 1 Nested Algorithm 1

procedure Foo2(Integer n)
    for $i \leftarrow 1; i \leq n; i \leftarrow 2 * i$ **do**
        for $j \leftarrow 1; j \leq n; j \leftarrow 2 * j$ **do**
            print "Hi"

Answer. First I will analyze the inner loop and then go to the outer loop.

So for the inner loop the initilization of j to be 1 takes 1 unit of time. Let K be the amount of times the loop runs until it terminates. This loop terminates when $K = (\log_2 n) + 1$. at each iteration, the loop does:

– The comparison $j \leq n$ takes 1 step.
– The update $j = j * 2$ takes 2 steps: one step to evaluate $j * 2$ and one step for the assignment.
– The body of the loop consists of a single print statement, which takes 1 step.

So the runtime complexity of the inner loop is:

$$1 + \sum_{j=1}^{(\log_2 n)+1} (1 + 2 + 1) = 1 + 4((\log_2 n) + 1) = 5 + 4 \log_2 n$$

Now for the outer loop:

Initialization of i takes 1 unit of time. this loop also runs $(\log_2 n) + 1$ times. at each iteration: The comparison of i to n takes 1 unit of time The evaluation and assignment of $i = 2 * 1$ takes 2 units of time the body takes $5 + 4 \log_2 n$ time since the body is just the inner loop.

So in total we get:

$$1 + \sum_{i=1}^{(\log_2 n)+1} (1 + 2 + 5 + 4 \log_2 n) = 1 + \sum_{i=1}^{(\log_2 n)+1} (8) + \sum_{i=1}^{(\log_2 n)+1} (4 \log_2 n) = 1 + 8 \log_2 n + 8 + 4(\log_2 n)^2 + 4 \log_2 n$$

We know that each loop will run $\log n$ times and that the total complexity is just multiplying those together so we can determine that $T(n) \in \theta((\log n)^2)$

□

4 Standard 15- Analyzing Code II: (Dependent 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.

Algorithm 2 Nested Algorithm 2

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

Answer. I will first analyze the inside loop the initialization of j takes 1 step. the loop will run i times At each iteration: comparing j to i takes a single step, update and assign of j takes 2 steps, and the body consists only of a print statement that takes 1 step So our formula for the inner loop is:

$$1 + \sum_{j=1}^{i}(1 + 2 + 1) = 1 + 4(i)$$

Now for the outer loop: the initialization of i takes one step. The loop will run 2n times. Then at each iteration it will take 1 step for comparison of i to 2n, update and assign take 2 steps, and the body of the loop takes $1 + 4i$ steps as found above. This brings the total steps to:

$$1 + \sum_{i=1}^{2n}(1 + 2 + 1 + 4i) = 1 + \sum_{i=1}^{2n}(4) + 4 * \sum_{i=1}^{2n}(i) = 1 + 8n + 4 * (n(2n + 1) = 1 + 8n + 8n^2 + 4n = 1 + 12n + 8n^2$$

We can now take our found run-time to determine the tight asymptomatic bound for $T(n)$. We can find the limit of $T(n)$ over n^2 to determine if that can be the functions bound.

$$\lim_{n\to\infty} \frac{1 + 12n + 8n^2}{n^2}$$

L'Hop

$$\lim_{n\to\infty} \frac{12 + 16n}{2n}$$

L'Hop

$$\lim_{n\to\infty} \frac{16}{2} = 8$$

I found the limit to be 8, meaning that $T(n) \in \theta(n^2)$.

□