Problem Set 4: The Countess of Functions

Due: **4:19pm Friday, 18 October 2019**

The purpose of this assignment is to continue to develop your understanding of the asymptotic operators, and to understand deeply the results in Chapter 5 of the textbook.

Collaboration Policy: You should do this assignment by yourself and submit your own answers. You may discuss the problems with anyone you want and it is also fine to get help from anyone on problems with LaTeX or Jupyter/Python. You are permitted to use any resources you find for this assignment. You should note in the *Collaborators and Resources* box below the people you collaborated with and any external resources you used (you may exclude resources you used exclusively for help with LaTeX or Jupyter/Python.

Collaborators and Resources: TODO: replace this with your collaborators and resources (if you did not have any, replace this with *None*)

This problem set includes both PDF and Jupyter notebook components. You should complete the answers to the PDF part by writing your answers in `ps4.tex`, and submitting your generated PDF file in collab.

As with previous problem sets, the first thing you should do in ps4.tex is set up your name as the author of the submission by replacing the line, `\submitter{TODO: your name}`, with your name and UVA id, e.g., `\submitter{Margaret Hamilton (mhh01z)}`. Before submitting your PDF, also remember to (1) list your collaborators and resources, replacing the TODO in `\collaborators{TODO: replace ...}`, and (2) replace the second line in ps4.tex, `\usepackage{uvatoc}` with `\usepackage[response]{uvatoc}` so the directions do not appear in your final PDF.

Asymptotic Operators

For all of these questions (and throughout cs3102), you should use the formal definitions of O, Θ, and Ω from Class 9.

Problem 1 For each sub-problem, indicate if the statement is *true* or *false* and support your answer with a convincing argument.

(a) $17n \in O(723n + \log n)$

(b) $\min(n^n, 3012) \in O(1)$

(c) $n^2 \in \Theta(n^3)$

(d) $2.0001^n \in O(2^n)$

(e) $log_n 10 \in \Theta(log_{2n} 17)$

Problem 2 Are there any two functions, $f, g : \mathbb{N} \to \mathbb{R}$ such that $f \notin \Omega(g)$ and $g \notin \Omega(f)$?

A good answer will either show a specific counter-example, or prove that no such two functions exist. A solid argument will need to use the formal definition of Ω, instead of relying on intuition (which is often counter-intuitive).

Soft-O and Little-o

Logarithms grow so slowly, they are practically "constants" — $\log_2 1\text{Trillion} < 30$. So, for any size problem we could compute on a real machine, theoreticians (and students who don't like to worry about manipulating logarithms) shouldn't waste their time worrying about logarithmic factors. Indeed, even polynomials on logarithms (i.e., $a_k(\log n)^k$ for any constant k) grow so slowly to usually be irrelevant. For this reason, we often use the "Soft-O" notation, $\widetilde{O}$:

Definition 1 ($\widetilde{O}$) *A function $f(n) : \mathbb{N} \to \mathbb{R}$ is in $\widetilde{O}(g(n))$ for any function $g(n) : \mathbb{N} \to \mathbb{R}$ if and only if $f(n) \in O(g(n) \cdot \log^k g(n))$ for some $k \in \mathbb{N}$.*

(Note: for convenience, we write $\log^k x$ to mean $(\log x)^k$. Also, we have seen the (constant) base of a $\log$ doesn't matter within our asymptotic operators, but if it is disturbing to have a $\log$ with uncertain base, it is fine to assume it is base 2.)

Problem 3 For each sub-problem, indicate if the statement is *true* or *false* and support your answer with a convincing argument.

(a) $n^2 \log n^3 \in \widetilde{O}(n^2)$

(b) $2.0001^n \in \widetilde{O}(2^n)$

(c) maximum number of comparison operations needed to sort a list of n items $\in \widetilde{O}(n)$

(Hint: by understanding the definition of $\widetilde{O}$ above, you should realize that one way to prove a function is in a $\widetilde{O}$ set is to choose a value for k used in the definition, but to disprove inclusion in $\widetilde{O}$ you need to show that there is no k that works.)

Another useful notation is "little-o" which is designed to capture the notion that a function g grows much faster than f:

Definition 2 (o) *A function $f(n) : \mathbb{N} \to \mathbb{R}$ is in $o(g(n)$ for any function $g(n) : \mathbb{N} \to \mathbb{R}$ if and only if for every positive constant c, there exists an $n_0 \in \mathbb{N}$ such that:*

$$\forall n > n_0 . f(n) \leq cg(n).$$

Problem 4 *Goldilocks and the Three Os*

(a) Prove that for any function f, $f \notin o(f)$.

(b) Prove that $n \log n \in o(n)$.

(c) ($\star$) Prove that for any $k \in \mathbb{N}$, $\log^k n \in o(n^\epsilon)$ for any $\epsilon > 0$.

(d) ($\star$) Prove that for any two functions f, g and constant $\alpha > 0$, $f \in \widetilde{O}(g) \implies g^{1+\alpha} \in \Omega(f)$.

Counting Circuits and Functions

Problem 5 *Equal to Constant Function* (TCS Exercise 5.3)

For every $k \in \mathbb{N}$ and $x' \in \{0,1\}^k$, show that there is an $O(k)$ line *NAND-CIRC* program that computes the function $EQUALS_{x'} : \{0,1\}^k \rightarrow \{0,1\}$ that on input $x \in \{0,1\}^k$ outputs 1 if and only if $x = x'$.

Problem 6 *Counting lower bound for multibit functions* (TCS Exercise 5.4)

Prove that there exists a number $\delta > 0$ such that for every n, m there exists a function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ that requires at least $\delta m \cdot 2^n/n$ NAND gates to compute. (If you are stuck, see this exercise in the book for a hint.)

Problem 7 *Random Functions are Hard* (TCS Exercise 5.8)

Suppose $n > 1000$ and that we choose a function $F : \{0,1\}^n \rightarrow \{0,1\}$ at random, choosing for every $x \in \{0,1\}^n$ the value $F(x)$ to be the result of tossing an independent unbiased coin. Prove that the probability that there is a $2^n/(1000n)$ line program that computes F is at most 2^{-100}. (If you are stuck, see this exercise in the book for a hint.)

End of Problem Set 4 (PDF part)
Don't forget to also complete and submit the Jupyter notebook!

