CS 321 - Algorithms - Fall 2019 | *Tyler Bates, Esteban Lopez*

Assignment 2

Due BEFORE 8:00AM on Monday 9/16/2019

On time / 20% off / no credit

Total points: 45

You are allowed to work with a partner on this assignment. If you decide to form a pair, make sure to include both names above, but submit only one file to Canvas.

This assignment will test your knowledge of asymptotic notation. You must write up your solutions to this assignment IN THIS FILE using LaTeX by filling in all of the boxes below. If your submitted `.tex` file does not compile, then you will receive 0 points. You should NOT add any LaTeX packages to your `.tex` file.

Make to sure to reread ALL of the requirements given in the handout for A1 pertaining to the structure and format of your proofs, since they apply to this assignment as well.

Submission procedure:

1. Complete this file, called `a2.tex`, with your full name(s) above and answers typed up below.
2. Compile this file to produce a file called `a2.pdf`. Make sure that this file compiles properly and that its contents and appearance meet the requirements described in this handout.
3. Create a directory called `a2` and copy exactly two files into this directory, namely:
   - `a2.tex` (this file with all of your answers and name(s) added)
   - `a2.pdf` (the compiled version of the file above)
4. Zip up this directory to yield a file called `a2.zip`
5. Submit this zip file to the Canvas dropbox for A2 before the deadline above.
6. Submit a single-sided, hard copy of your `a2.pdf` file BEFORE the beginning of class on the due date above.

Problem statements

1. **(5 points) Prove that $50 \cdot \log_3 N^{99} = O(\log_{27} N)$. Your proof MUST use the definition on slide 2-1. For full credit, your proof must use the smallest possible integer value for N_0 and must spell out the values of all required constants.**

   Proof:

   $F(n) = 50 \cdot log_3(N^{99}) = O(log_{27}N)$
   It suffices to show There exist positive constants c and N_0 such that:

for all N, if N N_0, then $f(N) \leq cg(N)$
$F(N) \leq 14850 \cdot log_{27}(N)$ For $N \geq 1$

1	$\forall N \geq 1,\ N = N$	Premise
2	$\log_3 N = \log_3 N$	Took the log base 3 of N on both sides
3	$14850 \cdot \log_3 N = 14850 \cdot \log_3 N$	Multiplied both sides by 14850
4	$14850 \cdot \log_3 N \leq 14850 \cdot \log_3 N$	By transitivity the LHS is $\leq$ the RHS
5	$\frac{14850 \cdot \log_3 N}{\log_3 27} \leq \frac{14850 \cdot \log_3 N}{\log_3 27}$	Divide both sides by $\log_3 27$
6	$4950 \cdot \log_3 \leq \frac{14850 \cdot \log_3 N}{\log_3 27}$	Simplified the LHS
7	$4950 \cdot \log_3 \leq 14850 \cdot log_{27} N$	Converted bases on RHS using a property of logs
8	$50 \cdot 99 \cdot \log_3 N \leq 14850 \cdot log_{27} N$	Factored out 50 from 4950
9	$50 \cdot \log_3 N^{99} \leq 14850 \cdot log_{27} N$	Converted $log_3 N$ to $\log_3 N^9 9$ using properties of logs $\square$

2. **(10 points) Prove that $8N^2 - 16N + 24 = \Theta(N^2)$. Your proof MUST use:**
   (a) the definition of the $\Theta(.)$ notation on slide 2-5 (and thus also those on slides 2-1 and 2-3), and
   (b) the constants defined by the formulas given at the bottom of page 46 in our text that apply to all quadratic functions.

Proof:

In order for f(N) to be $\Theta(N^2)$ it must be true that $c_1 \cdot N^2 \leq f(N) \leq c_2 \cdot N^2$ for some constants c_1 and c_2 $\forall N \geq N_0$. The values of c_1, c_2, and N_0 can be calculated because f(N) is a quadratic formula.

$$\begin{aligned} c_1 &= 8/4 \\ c_1 &= 2 \end{aligned}$$

$$\begin{aligned} c_2 &= 7(8)/4 \\ c_2 &= 56/4 \\ c_2 &= 14 \end{aligned}$$

$$\begin{aligned} N_0 &= 2 \cdot max(16/8, \sqrt{24/8} \\ N_0 &= 2 \cdot max(2) \\ N_0 &= 4 \end{aligned}$$

Subproof for $f(N) = \Omega(N^2)$

$f(N) \geq c_1 \cdot g(N)$

1	$N \geq 4$	4 is ourN_0 value
2	$N - 16 \geq -12$	Subtract 16 from both sides
3	$N - 16 \geq -24$	-24 is even smaller than -12
4	$N^2 - 16 \geq -24$	Transitivity
5	$N^2 - 16N + 24 \geq 0$	Add 24 to both sides
6	$8N^2 - 16N + 24 \geq 7N^2$	Add $7N^2$ to both sides
7	$8N^2 - 16N + 24 \geq 2N^2$	Transitivity

Subproof for $f(N) = O(N^2)$

$f(N) \leq c_2 \cdot g(N)$

1	$N \geq 4$	4 is our N_0 value
2	$N + 20 \geq 24$	Add 20 to both sides
3	$N^2 + 20N \geq 24$	Transitivity
4	$N^2 + 20N - 24 \geq 0$	Subtract 24 from both sides
5	$0 \geq -N^2 - 20N + 24$	Subtract LHS and add to RHS
6	$4N \geq -N^2 - 16N + 24$	Add 4N to both sides
7	$4N^2 \geq -N^2 - 16N + 24$	Squared LHS using transitivity
8	$13N^2 \geq 8N^2 - 16N + 24$	Add $9N^2$ to both sides
9	$14N^2 \geq 8N^2 - 16N + 24$	Transitivity

Since we are able to prove $f(N)$ is both $O(N^2)$ and $\Omega(N^2)$, we can conclude that $f(N)$ is in fact $\Theta(N^2)$. $\square$

3. **(10 points) Prove or disprove $4^N = \Theta(N^N)$. For full credit, your proof MUST use the definition on slide 2-5 or its negation. In other words, you must specify the required constant(s).**

Proof:

For 4^N to be $\Theta(N^N)$, 4^N must be $\Omega(N^N)$ and $O(N^N)$, by definition. This allows us to assume $4^N = \Omega$ is true. By definition of Ω, there is some positive constants $c > 0$ and $N_0 > 0$ such that $f(N) \geq c \cdot g(N)$. $f(N) = 4^N$, $g(N) = N^N$, then we know:

1	$4^N \geq c \cdot N^N$	By definition of Ω
2	$\frac{4^N}{N^N} \geq c$	Divide both sides by N^N
3	$(\frac{4}{N})^N \geq c$	Simplified LHS

This results in a contradiction because there is some $N > N_0$ such that $(\frac{4}{n}) < c$. $\square$

4. **(10 points) Prove or disprove $4^{2\log_3 N} = o(N^3)$. For full credit, your proof MUST use the formal definition on slide 3-5 or its negation. In other words, you must specify the required constant(s).**

Proof:

$f(N)$ is $o(N^3)$ if $\exists N_0$ such that $\forall$ constants c and $N \geq N_0$, we have $0 \leq f(N) \leq c \cdot g(N)$. Let $n_0 = c^{\frac{1}{2\cdot(\log_3 4)-3}}$.

1	$f(N) = 4^{2\cdot\log_3 N}$	Equation given
2	$f(N) = N^{2\cdot\log_3 4}$	Rewritten using property of logs
3	$f(N) = c$	By definition of c
4	$f(N) \leq c \cdot N^3$	Transitivity $\square$

5. **(10 points) Given a list of functions, your goal is to order them according to their rate of growth, from smallest to largest. For example, if given the following functions:**
N^2 $\quad 10N \quad N^2 - 5N + 12 \quad N \quad N\log N \quad 2^N$
the correct answer would be the following table:

N	$10N$
$N\log N$	
N^2	$N^2 - 5N + 12$
2^N	

in which all functions in a given row are big-theta of each other and little-o of all functions in the following row (if any). The left-to-right order within each row is not significant.

For this problem, logs **are base 2 unless otherwise specified. You must build an appropriately sized table that shows the correct ordering of the following 12 functions:**

$\sqrt{N^{14} 2^{\log\log^2 N}}$	$N^{0.05}\log_5 N$	$N\log N$	$\log\log N$	$N^{\frac{5}{\log_5 N}}$	5^{5N}
$N^{\frac{N}{\log_5 N}}$	N^5	5^N	$5^{N\log N^5}$	$N^5\sqrt{N^5}$	5^{N+5}

$\log \log N$		
$N^{0.05} \log_5 N$		
$N \log N$		
$N^{\frac{5}{\log_5 N}}$		
N^5		
$N^5 \sqrt{N^5}$		
$\sqrt{N^{14} 2^{\log \log^2 N}}$		
5^{5N}	5^N	5^{N+5}
$5^{N \log N^5}$		
$N^{\frac{N}{\log_5 N}}$		