Spring 2024

CS 412 (Algorithms: Design and Analysis)

Weekly Challenge 01: Getting Started...

Announced: Friday, January 12, 2024. Deadline: Friday, January 19, 2024 (11:59 pm PST).
Total marks: 2.

Instructions: Submit *individually* your solution as PDF with the file name as your studentID.pdf; typeset in LaTeX. You must submit your solution on Canvas.

---

1. (1 point) **Formal definition of Big-Oh:** Welcome to CS 412! As the first weekly, recall the formal definition of big-Oh as:

   $f(n)$ is $O(g(n))$ if there exist positive numbers c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$. Let's try to find the constants c and n_0 for the time complexity of some algorithm characterized by a function $f(n) = 2n^2 + 3n + 1 = O(n^2)$

   We can find these constants algebraically but let's plot the running time of $cg(n)$ for different values of c and n_0. You may use the values for N (n_0) starting with 1, 2, 3, 4, and 5 (see the table below).

   Plot the function $f(n)$ together with $g(n)$ [with different values of c and n_0 with values of n_0 on the $X - axis$ and the corresponding values of $f(n) and c.g(n)$ on the $Y - axis$.

   What is the smallest value of c and n_0 when $f(n) = O(g(n))$? Briefly comment [in two to three sentences only] on any notable observations drawn from this little exercise.

c	≥ 6	$\geq 3\frac{3}{4}$	$\geq 3\frac{1}{9}$	$\geq 2\frac{13}{16}$	$\geq 2\frac{16}{25}$	...	$\rightarrow$	2
N	1	2	3	4	5	...	$\rightarrow$	∞

2. (1 point) In computer science, $lg(n)$ by default refers to log to the base 2. The log function appears frequently in algorithm analysis and often, we ignore the base when performing asymptotic analysis. Does base really matter in asymptotic analysis? Plot the function $lg(n)$ with bases $2, 7, 10, 100$ for 'large values' of n. Write your observations (in two to three sentences only).