CS305: Homework 1

Assigned: Thursday, February 8, 2024
Due: Sunday, February 18, 2024

- Do all seven of the following problems. **Show your work**. Homework submissions must be submitted to the Spring **as a pdf**.
- You may work with others on the homework, but you **MUST** acknowledge the people you worked with at the top of your homework submission. Do **not** look at the web or use any kind of AI for solutions to homework problems. Looking for solutions does **not** help your problem solving powers.

Problems

Note that the book separates things into exercises and problems with exercises appearing at the end of a section and problems (which tend to be longer) appearing at the end of the chapter.

1. For each, either prove (by the definition of O and Ω) or state that it isn't true
   (a) $2n^3 + 4n$ is $O(n^5)$
   (b) $2n^3 + 4n$ is $\Omega(n^3)$
   (c) $2n^3 + 4n$ is $O(n^3)$
2. Answer these with (Yes or No) or (True or False) and a sentence why. I am not asking you to prove these by the definitions of Θ or o etc. like in the previous problem.
   (a) Is $2n^3 + 4n \quad \Theta(n^3)$?
   (b) Can you say that n^3 is an asymptotically tight bound for $2n^3 - 4n$?
   (c) Is n^2 an asymptotic lower bound for $2n^3 - 4n$?
   (d) n^2 is $o(n^4)$ Note: little o, not big O
   (e) n^2 is $o(n^3)$
   (f) n^2 is $o(n^2)$
   (g) n^2 is $o(n)$
   (h) 4^{lgn} is $\Theta(n^2)$

3. Order the following functions in a vertical list. Put a function $f_1(n)$ above function $f_2(n)$ if $f_1(n)$ is $o(f2(n))$. Put functions that are Θ of each other on the same line, separated by a comma.

   - $500n^2$
   - $10n^2 + 3n + 5$
   - 2^n
   - 1000
   - $2nlgn + 400n + 3$
   - $nlog_3n$
   - 5^n
   - $6n^3 + 1000000n + 5$
   - $n^3 + n^2 + 5$
   - 6
   - $4\sqrt{n}$
   - $2n^2lgn$

4. Suppose algorithm A runs in $\Theta(n)$ time and algorithm B runs in $\Theta(nlgn)$ time. Answer True or False for each and give a one sentence reason.

   (a) There may be values of n for which algorithm A runs faster than algorithm B.

   (b) There may be values of n for which algorithm B runs faster than algorithm A.

   (c) There is some value of n, such that for all values of n larger than that value, algorithm A runs faster than algorithm B.

   (d) There is some value of n, such that for all values of n larger than that value, algorithm B runs faster than algorithm A.

   (e) There may be a value for n for which algorithms A and B take the same amount of time.

5. In the 4th edition of textbook, Problem 4-2 Parameter Passing Costs. Consider the 3 parameter-passing styles each for a. recursive binary search and b. mergesort. Your answer should consist of six recurrences total for the worst-case running times of the two algorithms with each of the three parameter-passing styles. Do NOT do c. matrix-multiply-recursive.

6. Prove the following by induction:

$$\text{For all } x \neq 1, \qquad \sum_{i=0}^{d} x^i = \frac{x^{d+1} - 1}{x - 1}$$

7. Determine asymptotically tight bounds for each of the following by the Master Theorem if possible (and show which case it is and why the case holds). If not possible by the Master Theorem, indicate why and try the recursion tree method.

   (a) $T(n) = 8T(n/2) + 5n^3 + n^2$
   (b) $T(n) = 9T(n/2) + 5n^3 + n^2$
   (c) $T(n) = T(n-1) + 1$
   (d) $T(n) = 7T(n/2) + n^2$
   (e) $T(n) = 7T(n/2) + n^3$

