ADS Tutorial 1

Instructor: Aris Filos-Ratsikas TA: Kat Molinet

September 25, 2024

Problem 1

Work out (but don't bother proving) for each pair of expressions below, whether A is $O(B)$, $\Omega(B)$, $\Theta(B)$ (it could be none of these). Assume $k \geq 1, \epsilon > 0, c > 1$ are all constants.

A	B	O	Ω	Θ
$\lg^k n$	n^ϵ			
n^k	c^n			
$\sqrt{n}$	$n^{\sin n}$			
2^n	$2^{n/2}$			
$n^{\ln m}$	$m^{\ln n}$			
$\lg(n!)$	$\lg(n^n)$			

(Taken from Cormen, Leiserson, Riverst, and Stein (CLRS), Problem 3-2.)

Problem 2

Solve the following recursive formulas <u>without</u> using the Master Theorem.

A. $T(n) \leq T(n/2) + 4$

B. $T(n) \leq T(n/2) + 5n$

C. $T(n) \leq T(n/2) + 3n^2$

D. $T(n) \leq \frac{3}{2}T(n/2) + 1$

Sort the obtained formulas in terms of their asymptotic order.

Problem 3

Provide a (as-tight-as possible) bound for the asymptotic memory requirements of the following algorithms.

A. The MergeSort algorithm.

B. The QuickSort algorithm where we take the element $A[n]$ as the pivot.

Problem 4

A *majority element* in an array of n numbers is one that appears more than $\lceil n/2 \rceil$ times. Design an algorithm that receives as input a *sorted* array A of integers and outputs YES if a majority element exists and NO otherwise. Present the algorithm in terms of pseudocode. The algorithm should run in (worst-case) time $O(\log n)$ and you should formally prove its asymptotic running time. For simplicity, you may ignore issues regarding whether numbers are divisible by 2 (the algorithms can be adjusted to account for that via the appropriate use of the $\lceil \cdot \rceil$ function).