

COMP523 Tutorial 1

Coordinator: Aris Filos-Ratsikas

Demonstrator: Michail Theofilatos

September 26, 2019

Exercises from the textbooks

J. Kleinberg and E. Tardos. Chapter 2, Exercises 1, 2, 3, 6(a), 6(b).

Problem 1

Sort the following functions according to the O (“big oh”) and o (“small oh”) order:

$\log n^{1/2}$, $\log(9n)$, $\log n^3$, $2^{\log n}$, $2^{3 \log n}$, $2^{\log(9n)}$, n^2 , $n \log n$.

Problem 2

Recall that 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).