Divide & Conquer
Problem Set 1 – CS 6515/4540 (Fall 2025)

This problem set is due on **Thursday, August 28th**. Submission is via Gradescope. Your solution must be a typed pdf (e.g. via LaTeX) – no handwritten solutions.

1 O-Notation

1.1 LLM Question

Use the following prompt in an LLM (like chatGPT/CoPilot/Gemini) and upload the transcript of your session.

"Act like an undergrad with a CS major who wants to understand Big O notation. Now interactively ask me 5 questions (i.e., one by one while waiting for my answers) to improve your (and my) understanding of Big O. In the end, give me all the correct answers and constructive feedback."

1.2 Question:

We are interested in finding *all* constants for which the following $O(\cdot)$ bounds hold.

1. For which constants $c, d > 0$ do we have $n^c = O(d^n)$? Prove your answer.
2. For which constants $c, d > 0$ do we have $(\log(n))^c = O(n^d)$? Prove your answer.

Remark: c, d do not need to be integers. In particular, they could be smaller than 1.

2 Divide & Conquer Algorithm

Given a length n array A (you may assume there are only non-negative entries), construct an algorithm that returns the maximum possible product of *contiguous* entries. For example when $A = [0.5, 2, 0.25, 2, 0.75, 2, 0.5]$, the maximum product is 3 from the sub-array $[2, 0.75, 2]$.

Problem: Construct a divide & conquer algorithm MaxProduct($A[1...n]$) that solves the above problem. Your answer must provide the following:

(a) A pseudo-code description of your algorithm.

(b) A correctness argument. You may provide a proof by induction, but answering the following questions suffices:

(a,i) Why is the base case correct (when A has length 1)?

(a,ii) Why does the algorithm return a correct answer, if it has the solution for the left and right half of the array, i.e., MaxProduct($A[1...n/2]$) and MaxProduct($A[n/2+1, ..., n]$)).

(c) Complexity analysis (e.g., via Tree Method, induction, or reusing a bound from class).

There exist iterative algorithms for this problem, but the purpose of this exercise is to practice divide & conquer. Your solution must be a divide & conquer algorithm, i.e., recursively solving the problem by splitting array $A[1...n]$ into $A[1...n/2]$ and $A[n/2+1...n]$.

For full points, your solution should be as fast as possible.

3 Median of Medians

The standard median of the median algorithm uses groups of five. Instead, consider a median of the median algorithm, with groups of $M \geq 3$.

1. Give the recurrence for this algorithm (with respect to M) to calculate the worst-case runtime.
2. Suppose $M = 7$, now solve the recurrence by induction. Make sure to clearly write the induction hypothesis, base case, and induction step.
3. Which of the following is true about the running time of the algorithm in part (b.) for $M = 7$ (as compared to $M = 5$ from class): (i) running time improves to $o(n)$, (ii) running time stays $\Theta(n)$, or (iii) running time worsens to $\omega(n)$? Briefly explain your choice.