Weekly Problems: 3

Due: 2pm on 27/08/25

Instructions

- Discussion is allowed and infact encouraged
- Answers must be written by yourself.
- All sources that are used to reach the solution must be mentioned.

---

(1) Recall the maximum subarray problem:

Input: Array of length n whose entries are all integers. [So, $A \in \mathbb{Z}^n$]

Output: A maximum sub-array of A.

That is, $i_0, j_0 \in [n]$ with $i_0 < j_0$ s.t.

$$\sum_{k=i_0}^{j_0} A[k] = \max_{i \leq j \in [n]} \left\{ \sum_{k=i}^{j} A[k] \right\}$$

a) Write a pseudocode for solving this problem that takes time $\Theta(n^2)$. Argue correctness.

Remember to give an instance and show why your algo takes time $\Omega(n^2)$ on that instance.

[6]

b) How would the algorithm change if it was allowed for j_0 to be equal to i_0?

[2]

(2) Guess a good upper bound for the recurrence

$$T(n) = T(n-1) + T(n/2) + n.$$

and prove the correctness of your guess using induction. [2+2].

(3) Assume $\mathrm{STRASSEN}_2(A, B)$ is an algorithm that takes as input two 2×2 matrices A, B (over $\mathbb{Q}$) and returns their product using '7' $\mathbb{Q}$-multiplications

Use this as a subroutine to write

the pseudocode for an algorithm that takes as input two $n \times n$ matrices (over $\mathbb{Q}$) A, B and returns their product using $n^{\log_2 7}$ $\mathbb{Q}$-multiplications. [4]

Assume that n is a power of 2 for simplicity

(4) Solve the following recurrences:

a) $T(n) = T(n-a) + T(a) + c \cdot n$

b) $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$

Here a, α, c are all constants s.t.

- $a \geq 1$
- $0 < \alpha < 1$
- $c > 0$

[3+3]

(5) What is the smallest possible depth of a leaf in a decision tree for a comparison sort? [2]

—