Name:

Student ID #:

- You are expected to work on this assignment on your own
- Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
- When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
- Always remember to analyze the time complexity of your algorithms
- Homework has to be submitted electronically on Gradescope by the deadline. No late assignments will be accepted
Problem 1. (10 points)

Give a divide-and-conquer algorithm for multiplying two polynomials of degree $n$ in time $O(n^{\log_2 3})$.

Answer:
Problem 2. (10 points)

For an $n$ that is a power of 2, the $n \times n$ Weirdo matrix $W_n$ is defined as follows. For $n = 1$, $W_1 = [1]$. For $n > 1$, $W_n$ is defined inductively by

$$W_n = \begin{bmatrix} W_{n/2} & -W_{n/2} \\ I_{n/2} & W_{n/2} \end{bmatrix},$$

where $I_k$ denotes the $k \times k$ identity matrix. For example,

$$W_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad W_4 = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad W_8 = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & 1 & -1 & 0 & -1 & 1 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Give $O(n \log n)$-time algorithm that computes the product $W_n \cdot \bar{x}$, where $\bar{x}$ is a vector of length $n$ and $n$ is a power of 2.

Answer:
Problem 3. (10 points)

Given an array $A$ of $n$ (possibly negative) integers, find two indices $1 \leq i \leq n$ and $1 \leq j \leq n$ such that the value of $\sum_{k=i}^{j} a_k$ is maximized.

Here some examples (the solution is underlined):

- $A = [-2, 11, -4, 13, -5, 2]$ which has answer $20$,

- $A = [1, -3, 4, -2, -1, 6]$ which has answer $7$,

- $A = [-1, 4, -3, 5, -2, -1, 2, 6, -21]$ which has answer $11$.

Write an $O(n \log n)$-time divide and conquer\(^1\) algorithm for the problem described above. The algorithm should return $i$ and $j$. If all elements of the array are negative, the algorithm should return $i = j = 0$.

Answer:

\(^1\)A $O(n)$ dynamic programming algorithm for this problem exists, but here you are supposed to give the slower divide and conquer algorithm.