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)
Answer the following two questions about Karatsuba’s divide-and-conquer integer multiplication algorithm:

1. First, generalize Karatsuba’s algorithm to multiply an \(n\)-bit integer by an \(m\)-bit integer, where \(n \geq m\), in \(O(nm^{0.5849...})\) time.

2. Second, consider this scenario. A student has been trying to speed-up Karatsuba’s divide-and-conquer integer multiplication algorithm. Given two numbers \(x, y\) with \(n\) bits each, her/his algorithm (1) first divides both \(x\) and \(y\) into four equal-length pieces, then (2) expresses the product \(x \cdot y\) using 4 multiplications of these \(n/4\)-bit pieces, followed by a merging step that takes \(\Theta(n^p)\) where \(p > 1\). What condition on \(p\) would give her/him a faster algorithm than the Karatsuba’s algorithm covered in class?

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,

\[
W_2 = \begin{bmatrix}
1 & -1 \\
1 & 1
\end{bmatrix}, \quad W_4 = \begin{bmatrix}
1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 \\
1 & 0 & 1 & -1 \\
0 & 1 & 1 & 1
\end{bmatrix}, \quad W_8 = \begin{bmatrix}
1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & 0 & 1 & -1 & -1 & 0 & 1 & -1 \\
0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\
1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\
0 & 1 & 0 & 0 & 1 & 1 & -1 & -1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & -1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

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 of numbers $X = \{x_1, x_2, \ldots, x_n\}$, an exchanged pair in $X$ is a pair $(x_i, x_j)$ such that $i < j$ and $x_i > x_j$. Note that an element $x_i$ can be part of up to $n - 1$ exchanged pairs, and that the maximal possible number of exchanged pairs in $X$ is $n(n - 1)/2$, which is achieved if the array is sorted in descending order. Give a divide-and-conquer algorithm that counts the number of exchanged pairs in $X$ in $O(n \log n)$ time.

Answer: