This assignment is due Wednesday, March 14 at 11pm via electronic submission. Collaboration is permitted, according to the rules specified in the syllabus.

Read Chapter 5.1–5.4 in Algorithm Design.

1. Suppose that \( T(n) \) is a function that satisfies the following recurrence:

\[
T(n) = \begin{cases} 
0 & \text{if } n = 1 \\
T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n \log_2 n & \text{if } n > 1
\end{cases}
\]

Prove from first principles that \( T(n) \) is \( \Theta \left( n \log^2 n \right) \) when \( n \) is a power of 2.

**Hint:** prove that \( T(n) = n/2 \times (\log_2 n) (1 + \log_2 n) \) when \( n \) is a power of 2.

2. Given a list of \( n \) integers \( A[1..n] \), your goal is to rearrange the integers to be in ascending order via a sequence of reversal operations: pick two indices \( i < j \) and reverse the sublist \( A[i..j] \). The cost of a reversal operation is \( j - i + 1 \)—the length of the sublist.

(a) Given a list \( A[1..n] \) containing only 0s and 1s, design a divide-and-conquer algorithm that sorts \( A \) via a sequence of reversal operations of \( O(n \log n) \) cost.

(b) Given a list \( A[1..n] \) of \( n \) integers, design a divide-and-conquer algorithm that sorts \( A \) via a sequence of reversal operations of \( O(n \log^2 n) \) cost.

**Hint:** use your algorithm from (a) as a subroutine and analyze it using Problem 4.1.

**Context:** sorting-by-reversals plays a key role in inferring the evolutionary relationship between two genomes.

3. Consider a database of \( n \) hotels in which each hotel has two fields of interest to a customer: the price of a standard room and the distance to the beach. We say that hotel \( i \) dominates hotel \( j \) if hotel \( i \) is both cheaper and closer to the beach than hotel \( j \). Design a divide-and-conquer algorithm to identify all hotels that are not dominated by any other hotel. In database terminology, this is known as a skyline query. The running time of your algorithm should be \( O(n \log n) \) in the worst case.