1. Show that the Master Theorem cannot be applied to the recursion \( T(n) = 2T(n/2) + n \log_2 n \).

2. Apply the fast multiplication algorithm to compute \( 3228 \times 5424 \). What is the exact number of computational steps the algorithm had to take on this input? How many multiplications and how many additions are executed? How many computational steps would we need if we used the naive way of multiplying polynomials (i.e., multiply every component with every component and then sum up)?

3. Find an ordering of the integers \( 1, 2, \ldots, 16 \), that when given as an input to the merge-sort algorithm produces the (a.) best (b.) worst-case running time on an input of this length. Produce the same list for the quick sort algorithm.

4. Sort the numbers \( \{14, 8, 7, 15, 2, 10, 9, 1\} \) in increasing order, using Batcher’s algorithm (the merge-sort type algorithm involving bitonic arrays) discussed in the lecture. What is the exact running time of the algorithm on this input? What is the worst-case running time of this algorithm on a list of \( n \) numbers? What is the worst-case running time of the regular merge-sort algorithm?

5. Let \( S = [-7, -4, -3, 0, 1, 2, 3, 5, 8, 11] \) be a sorted array of integers. Define an algorithm that, in \( O(n) \) time, either finds two elements in \( S \) that sum up to 0 or outputs that there are no such elements.