1. Analyze the time complexity of the following variant of merge sort: given a constant $k$, divide the array into $k$ parts, sort each part recursively, and merge the results.

2. Show how to apply the Strassen’s algorithm to multiply an $n^2 \times n$ matrix with an $n \times n^2$ matrix when $n$ is a power of 2. Analyze its time complexity.

3. Show how to apply the fast Fourier Transform to multiply three polynomials $A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$ of degree $n - 1$, $B(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{n-1} x^{n-1}$ of degree $n - 1$, and $C(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$ of degree $n - 1$ when $n$ is a power of 2. Analyze its time complexity.

4. Given an undirected graph with more than two vertices with unique edge weights between all pairs of vertices, prove that the edge with the largest weight cannot be in a minimum spanning tree.

5. Given a connected undirected graph with edge weights, design an algorithm to check if a given tree is a minimum spanning tree. Analyze its time complexity.