1. Present a CRCW PRAM algorithm for finding the maximum of \( n \) given numbers in \( O(1) \) time using \( n^{1+\epsilon} \) processors, where \( \epsilon \) is any constant > 0.

2. Input is a sequence of \( n \) numbers (not necessarily in sorted order). The problem is to compute the right neighbor of each element in the sorted order. For example if the input is 6, 12, 5, 3, 17, 11, the output will be 11, 17, 6, 5, \( \infty \), 12. Present a Las Vegas algorithm for this problem that runs in \( \tilde{O}(1) \) time. You can use up to \( n^2 \) CRCW PRAM processors.

3. Array \( A \) is an almost sorted array of \( n \) elements. It is given that the position of each key is at most a distance of \( d \) away from its final sorted position where \( d \) is a constant. Give an \( O(1) \) time \( n \)-processor EREW PRAM algorithm to sort \( A \). Prove the correctness of your algorithm using the zero-one principle.

4. Present an \( O(\log n) \) time algorithm to compute the FFT of a given vector of length \( n \). You can use up to \( n \) CREW PRAM processors. As a consequence present a parallel algorithm to compute the product of two given polynomials.

5. The array \( A \) is an array of \( n \) keys, where each key is an integer in the range \([1, n]\). The problem is to decide whether there are any repeated elements in \( A \). Show how you do this in \( O(1) \) time on an \( n \)-processor CRCW PRAM. Which version of the CRCW PRAM are you using?

6. Strassen’s algorithm for matrix multiplication was introduced in class. Using the same technique, design a divide-and-conquer algorithm for matrix multiplication that uses only \( n^{\log_27} \) CREW PRAM processors and runs in \( O(\log n) \) time.