1. Show that we can sort \( M^2 \) keys on the Parallel Disks Model in seven passes through the data (assuming that \( B = \sqrt{M} \)). \textit{Hint:} Use the LMM sort algorithm. How many passes will be needed to sort \( M^3 \) keys if we use the LMM algorithm (for the case of \( B = \sqrt{M} \))? 

2. Input are two \( n \times n \) matrices \( A \) and \( C \) residing in \( D \) disks. Present an algorithm for multiplying these matrices using \( O\left(\frac{n^3}{DB}\right) \) parallel I/O operations. To begin with these matrices are striped across the disks in a row-major order. Specifically, let \( R \) be any row of \( A \) or \( C \). The first \( B \) elements of \( R \) are in disk 1, the next \( B \) elements of \( R \) are in disk 2, etc., where \( B \) is the block size. Assume that \( M = \Theta(DB) = \Theta(n) \).

3. (Gusfield) Given a set \( S \) of \( k \) strings, we want to find every string in \( S \) that is a substring of some other string in \( S \). Assuming that the total length of all the strings is \( M \), give an \( O(M + k^2) \)-time algorithm to solve this problem.

4. (Gusfield) Give an algorithm to take in a set of \( k \) strings and to find the longest common substring of each of the \( \binom{k}{2} \) pairs of strings. Assume each string is of length \( n \). Since the longest common substring of any pair can be found in \( O(n) \) time, \( O(k^2 n) \) time clearly suffices. Now suppose that the string lengths are different but sum to \( M \). Show how to find all the longest common substrings in time \( O(kM) \).

5. Let \( T \) be a text of length \( m \). Assume that the suffix array and the LCP array have already been constructed for \( T \). Show how to identify all the occurrences of a pattern \( P \) in \( T \) in \( O(\log m) \) time. You can use up to \( n \) CRCW PRAM processors, where \( n = |P| \).