1. Given a set S of k strings with total length n, devise an O(n)-time algorithm to find every string in S that is a substring of another string in S.

2. Consider two strings A and B of lengths n and m, respectively. C is an *interleaving* of A and B if a subsequence of C forms A and the remaining characters not in the subsequence form B. For example, if A = abac and B = bbc, then a possible interleaving is abbabcc. Devise an efficient algorithm to determine whether a string C is an interleaving of A and B.

3. Let T be a string of length m and let \mathcal{S} be a multiset of n characters. Devise an O(m) time algorithm to find all substrings of T composed of all the characters from \mathcal{S} . For example, if $\mathcal{S} = \{a, b, c, c\}$ and T = acbcagaba, then cbca is a substring of T formed from the characters of \mathcal{S} .