1. Manber 5.11. State (and justify) the asymptotic running time of your algorithm, and justify why it is efficient.

2. Manber 5.18

3. Let \( \Sigma \) be an alphabet of symbols, and let \( X, Y, Z \in \Sigma^* \). Say that \( Z \) is a shuffle of \( X \) and \( Y \) if \( |Z| = |X| + |Y| \) and if \( X \) and \( Y \) occur as disjoint substrings of \( Z \). For example, if \( X = \text{close} \) and \( Y = \text{class} \), then \( \text{cloclasess} \), \( \text{classclose} \), and \( \text{ccllaossse} \) are all shuffles of \( X \) and \( Y \), but \( \text{clacloesss} \) and \( \text{classosecl} \) are not.

   Describe an efficient algorithm to determine whether \( Z \) is a shuffle of \( X \) and \( Y \). Let \( M \) be the length of \( X \) and \( N \) the length of \( Y \). What is the time complexity of your algorithm as a function of \( M \) and \( N \)?