Fundamental Algorithms, Section 003
Homework 8, Additional Problems, Fall 22.

1. In Roman times Latin was written without any spaces. Thus a text consisting of a sequence of words $w_1, w_2, \ldots, w_m$ was presented as the characters for each word appearing one after another.

In this problem you are given a string $s = s_1 s_2 \ldots s_n$ where each s_i is a single character from a constant sized set (or alphabet) Σ. You are also provided with a dictionary D. Given any substring $s_{ij} = s_i s_{i+1} \ldots s_j$, $D(s, i, j) =$ True if s_{ij} is a known word and False otherwise; D reports its result in $O(1)$ time. Your task is to give a polynomial time algorithm to determine whether s is string of known words or not.

a. Give a recursive formulation of a solution to this problem, explaining briefly why it is correct.

b. What is the running time of an efficient recursive implementation? Justify your solution.

c. What additional information would you need to determine the known words forming s when s is formed from a sequence of known words.

2. Consider the following game. It consists of a sequence of n cards $C_1, C_2, \ldots, C_n$, laid out in this order. Card C_i has known value v_i.

There are two players, A and B. The players alternate turns. On their turn, the players picks one of the two cards at the current ends of the sequence. Player A wins the total value of her cards minus the total value of B's cards, and symmetrically for Player B. Player A goes first.

a. Give a recursive formulation of a solution to this problem, explaining briefly why it is correct.

b. What is the running time of an efficient recursive implementation? Justify your solution.

c. What additional information would you need to determine a winning strategy for player A; i.e. following precomputation, it must report each move Player A should make following each possible move by Player B.