

TA Lab Week 10 – Strings

The goal of this lab is to explore further the structure of strings and associated applications.

Exercise 0 (preliminaries) – Prefix

Given a 'sentence' that consists of a sequence of words separated by a single space, and a 'searchWord', check if 'searchWord' is a prefix of any word in the sentence. Return the index of the word in sentence where 'searchWord' is a prefix of this word (1-indexed). If 'searchWord' is a prefix of more than one word, return the index of the first word (minimum index). If there is no such word return -1.

Hint

A prefix of a string S is any leading contiguous substring of S.

Exercise 1 – Autocomplete

Build a general autocomplete engine for words or, even better, sentences. This engine takes in a database (*) and a query (or list of queries), then returns the autocompleted query based on lexicographic order of matches.

Build another autocomplete engine, this time returning matches in descending order of weights, where weights represent the frequency of historical searches.

(*) For simplicity, design the database as a simple list of sentences (or words). For the second engine, the database can be a list of tuples, where the first element is a sentence/word and the second its associated weight.

Exercise 2 – No Prefix Set

Imagine you are given N strings, with each string containing only lowercase letters from a-j (both inclusive). The set of N strings is said to be a "good" set if no string is prefix of any other string, or else it is said to be "bad". If two strings are identical, they are considered prefixes of each other.

For example: {aab, abcde, aabcd} is a "bad" set because aab is prefix of aabcd.

Design and implement an algorithm that, given in input a set of strings, determines whether the set is good or bad; if it is bad, the algorithm should also return the first string for which the condition fails.