CSE 202: Design and Analysis of Algorithms **(Due: 10/26/19)**

Homework #2

Instructor: Ramamohan Paturi *Name:* Shihan Ran, *Netid:* A53313589

Problem 1: Maximum Length Chain of Subwords

Problem Description

Input: We are given a set of n distinct strings of length at most k over a finite alphabet σ.

Output: A sequence of strings that form a chain under the (consecutive) subword relation; i.e., if the output is $w_1, w_2, \ldots, w_t$ then we can write $w_{i+1} = uw_iv$ for some strings u, v.

Find a chain of maximum length.

Solution

(High-level description)

We use dynamic programming to solve this problem.

Let $T(n)$ be the maximum length of chain containing w_n. Then we can have the state transition function as follows:

$$T(n) = \max(T(i) + 1), \text{ where } w_n = uw_iv, 1 \leq i < n$$

Since our algorithm should output a sequence of string chain. We need to store some additional values:

1. $maxIndex$: the tail string's index of the maximum length chain
2. $P(n)$: the index i that maximize $T(n)$

With $maxIndex$ and $P(n)$, we can backtrack our maximum length chain of subwords following the order

$$[w_{maxindex}, w_{P(maxindex)}, w_{P(P(maxindex))}, \ldots]$$

To find out whether $w_n = uw_iv$, we need to define a new algorithm called $isSubsequence$. We can choose a pattern matching algorithm to do the string comparison. Here, we choose KMP algorithm and the time complexity is proportional to the length of the string, which is O(k).

(Pseudo Code)

(Correctness)

(Time complexity)

The $T(n)$ loop will traverse n strings, which takes $O(n)$ time. At each $T(n)$, we will iterate through $i, 1 \leq i < n$ elements, takes $O(n)$ time. When at each w_i, we do a KMP algorithm to find out whether w_i is a subword of w_n, which will take O(k) time, and finding $\max(T(i) + 1)$ will only take $O(1)$.

Overall, our algorithm takes $O(kn^2)$.

Problem 2: Business plan

Problem Description

Solution

(High-level description)

(Pseudo Code)

(Correctness)

(Time complexity)

Problem 3: Minimum Cost Sum

Problem Description

Solution

(High-level description)

(Pseudo Code)

(Correctness)

(Time complexity)

Problem 4: Speech recognition

Problem Description

Solution

(High-level description)

(Pseudo Code)

(Correctness)

(Time complexity)