Algorithm Design and Analysis
Assignment 4
Deadline: May 26, 2024

1. (25 points) Design a polynomial time algorithm to find the longest palindrome that is a subsequence of a given input string. Please refer to the last slide of Lecture 11 for the definition of palindrome.

2. (25 points) In the class, we have seen a dynamic programming algorithm for computing the edit distance between strings of length m and n creates a table of size $n \times m$ and therefore needs $O(mn)$ space. Show how we can reduce it to linear space.

3. (25 points) Two strings $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_m$ are given as inputs.

   (a) Design an $O(mn)$ time algorithm that decides the length of the *longest common substring*, i.e., the largest k for which there are indices i and j with $x_i x_{i+1} \cdots x_{i+k-1} = y_j y_{j+1} \cdots y_{j+k-1}$.

   (b) Design an $O(mn)$ time algorithm that decides the length of the *longest common subsequence*, i.e., the largest k for which there are indices $i_1 < i_2 < \cdots < i_k$ and $j_1 < j_2 < \cdots < j_k$ with $x_{i_1} x_{i_2} \cdots x_{i_k} = y_{j_1} y_{j_2} \cdots y_{j_k}$.

4. (25 points) In the *subset-sum problem*, you are given a set $T = \{a_1, \ldots, a_n\}$ of n positive integers and a positive integer k as inputs, and you are to decide if there is a subset S with sum exactly k. Notice that a set in this problem may contain multiple copies of an integer.

   (a) Design an $O(kn)$ time algorithm for this problem. Note: This is not a polynomial time algorithm. In fact, as I remarked in the class, the subset-sum problem is a well-known NP-complete problem that we do not believe to be solvable in polynomial time.

   (b) Suppose now you are guaranteed that there exists a subset S with sum exactly k and you are given an extra input parameter $\varepsilon > 0$. Design an algorithm to find a subset S' such that

   $$\sum_{a_i \in S'} a_i \in [(1-\varepsilon)k, (1+\varepsilon)k] \,.$$

   Your algorithm's running time should be polynomial in terms of $1/\varepsilon$ and n. Prove the correctness of your algorithm, and analyze its running time.

5. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.