

CSE 3500 – Algorithms and Complexity
Homework 8

Question 1(40 points)

In class, we studied the problem of finding the longest common subsequence (LCS) of two sequences X and Y (with n and m elements respectively).

Recall that we have a two-dimensional array $LCS[i, j]$ for X and Y where $0 \leq i \leq n$, $0 \leq j \leq m$. Here, $LCS[i, j]$ is equal to the length of LCS of $X[1..i]$ and $Y[1..j]$. Now we let $S_1 = \text{ABAABBA}$ and $S_2 = \text{BAAABAB}$. Here are your tasks.

1. (20 points) First, create and fill in the dynamic programming table $LCS[n, m]$.
2. (10 points) Then, use the table to find the length of LCS between the two sequences.
3. (10 points) Finally, use the table to find the LCS itself (instead of just the length). Note: you need to show how you get the LCS: you need to mark the letters in S_1 and S_2 in the LCS while doing the traceback.

Question 2 (30 points)

Palindrome is a nonempty string over some alphabet that reads the same forward and backward. Examples of palindromes are `civic`, `racecar`, and `aibohphobia` (fear of palindromes).

Write an algorithm using Dynamic programming approach to find the longest palindrome subsequence. For example, given the input `character`, your algorithm should return 5, because the longest palindrome subsequence is `carac`.

1. (5 points) **First explain your high-level idea.**
2. (10 points) **Write the recurrence relation for your algorithm?**
3. (10 points) ***Write the algorithm***
4. (5 points) ***What is the time complexity of your algorithm***

Question 3 (30 points)

Imagine a game where players earn different points with each move. There's also a target score to reach. Our goal is to determine the number of possible ways to reach that target score using the given points.

Example 1: Input: points = {3,5,10}, target score n = 20 Output: 4 There are following 4 ways to reach 20 (10, 10) (5, 5, 10) (5, 5, 5, 5) (3, 3, 3, 3, 3, 5)	Example 2: Input: n = 13 Output: 2 There are following 2 ways to reach 13 (3, 5, 5) (3, 10)
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------

1. (10 points) **First explain your high-level idea.**
2. (10 points) ***Write the algorithm***
3. (10 points) ***What is the time complexity of your algorithm***