Answer the questions below through the Udacity site.

Note that for some questions it is not sufficient to simply pass the tests on the site. Your code must have a certain running time, and in some cases you must also answer additional questions in the comments of your code.

1. Suppose there is a procedure that decides a language in time, where n is the length of the input string. Use dynamic programming (not memoization) to create an algorithm that decides in time. Implement your strategy in python here.

<https://www.udacity.com/course/viewer#!/c-ud557/l-1209378918/e-3160768618/m-3234618854>

1. Consider an arbitrary string . A *subsequence* of is a string of the form where . For example ‘ape’ is a subsequence of ‘apple.’ A string is *palindromic* if it is equal to its own reverse (i.e. the string is the same whether read backwards or forwards).

We will use dynamic programming to find the length of the longest palindromic subsequence of . For let be the length of the longest palindromic subsequence of the substring (the substring going from the ith character to the jth character of x, not including the jth character).

* 1. Derive a recurrence for L(i, j), remembering to include the base case?
  2. Why is your recurrence correct?
  3. Write a dynamic programming algorithm based on the recurrence for L(i, j) in python. Use backtracking to construct a longest palindromic subsequence.
  4. What is its running time?

Answer the above questions in the comments of your submission to Udacity at

<https://www.udacity.com/course/viewer#!/c-ud557/l-1209378918/m-2549628613>

1. Let and let n be two polynomials. Recall that . Implement an for finding the coefficients of at

<https://www.udacity.com/course/viewer#!/c-ud557/l-1209378918/m-2697878553>

Explain your running time analysis in your comments. It’s okay if its faster than .

(Bonus 10pts) Make your algorithm . Explain your running time analysis in your comments.

1. Given two sets and of integers, their sum set is defined to be For each , let denote the number of ways in which c can be obtained, i.e., For example, if and , then their sum set is and , , , , , and = 1.
   1. For two arbitrary sets and of integers, what is the worst-case complexity of computing their sum set?
   2. Suppose now that consist of integers in the range . Give an upper bound on the number of distinct elements in their sum set.
   3. With the above assumption, give an to find for each

Answer the above questions in the comments of your submission to Udacity at

<https://www.udacity.com/course/viewer#!/c-ud557/l-1209378918/m-2569679056>