**程式作業2**

111504504　楊愷晴

**一、程式碼**

**Way1: dynamic programming (Python)**

A white background with black and red text

Description automatically generated

利用**動態規劃**的精神，c[i][j]表示str1[1:i]和str2[1:j]的LCS長度，從c[1][1]開始找，最右下角的c[m][n]即為所求。利用兩層for迴圈計算完table c[][]。**Time complexity = O(n^2)**。

**Way2: brute force (Python)**

A white background with black text

Description automatically generated

從兩個字串的最後一個字元開始，一樣的話就去找str1[0:m-1]和str2[0:n-1]的LCS（去掉兩邊最後一個字元後繼續找）；不一樣的話就找Max{ str1去掉最後一個字元、str2不變的LCS，str1不變、str2去掉最後一個字元的LCS}。若其中一個string跑到index -1就終止遞迴。

**Time complexity = O (2^n)**。

**二、實驗測試兩種方法**

測試環境

CPU: 13th Gen Intel® Core™ i7-1360P

Operating System: Windows 11

Python3.12

產生測資

手動產生length=3~10的英文字母字串，每種長度各5筆，存在’input.txt’，讀取該檔案，用Python time module 取得lcs\_brute和lcs\_len兩個函式計算所需的時間，並寫到’output.txt’（在google colab執行）。

寫一個plot.py，讀取’output.txt’，用matplotlib畫圖（在google colab執行）。

繪圖結果

A graph with a line going up

Description automatically generated

根據圖表，大概在n=5後，dp方法所需的時間明顯小於brute force。