

Homework 6

109610025 陳品妍

Result:

```
please enter array size of x : 7
please input a sequence x : 1 2 3 2 4 1 2
please enter array size of y : 6
please input a sequence y : 2 4 3 1 2 1
please enter array size of z : 7
please input a sequence z : 1 2 1 3 2 4 1
The length of the LCS is: 4
sequence w is:
2      3      2      1
```

Discussion:

窮舉法為找出所有組合並檢查每一個組合是否為 longest common sequence。以題目的長度來看：三個輸入序列(x,y,z)的長度分別為 x_length 、 y_length 、 z_length ，那 x 的所有可能子序列則有 2^{x_length} 種組合，對於 y 以及 z 個別為 2^{y_length} 、 2^{z_length} 。接著要檢查子序列是否為 longest common sequence，因此每個挑選的 xyz 子序列比較最多涉及 $O(x_length + y_length + z_length)$ 次。因此，總體而言的時間複雜度

$$O(2^{x_length} * 2^{y_length} * 2^{z_length} * (x_length + y_length + z_length))$$

相當龐大的時間複雜度，尤其是當序列很長的時候，時間成本將非常大。

相反地 LCS 運用 dynamic programming 來降低時間複雜度，首先藉由 b 矩陣(三維矩陣 $(x_length + 1) * (y_length + 1) * (z_length + 1)$) 來紀錄 $x[i]y[j]z[k]$ 的 lcs 是從哪一個前面序列 ($x[i - 1]y[j]z[k]$, $x[i]y[j - 1]z[k]$ 或 $x[i]y[j]z[k - 1]$) 而來，功用主要是在幫助我們重建 lcs sequence 的答案。接下來也需要一個跟 b 一樣大小的三維矩陣 c 來紀錄 longest common sequence 的長度。接著使用三層迴圈來檢查所有子序列為一的組合，如果當前數字相同，根據課本的 solution 推導， $c[i][j][k] = c[i - 1][j - 1][k - 1] + 1$ (left - up - back) 否則

$$c[i][j][k] = \max(c[i - 1][j][k], c[i][j - 1][k], c[i][j][k - 1])$$

同時也要記錄答案的來源:

$c[i - 1][j][k]$ represents answer from $i - 1$ direction

$c[i][j - 1][k]$ represents answer from $j - 1$ direction

$c[i][j][k - 1]$ represents answer from $k - 1$ direction

因此可以幫助我們在重建答案時，知道答案在哪個方向。這樣的演算法時間複雜度為

$$O(x_length * y_length * z_length)$$

其中來自了三層迴圈的時間複雜度。這樣的複雜度使得在面對資料較大的問

題，時間不至於像窮舉法遽增。最後再將答案印出來，從最後一個開始檢查 b 如果 $b == 1$ 則在迴圈跳回來的時候直接印 $x[i]$ ，中間牽涉多層答案的方向($b == 2$ 答案來自 $i-1$ ， $b=3$ 答案來自 $j-1$ ， $b==4$ 答案來自 $k-1$)，再一一進行遞迴，之後就可以得到答案了。