HW2　Longest palindrome subsequence problem

B10532011 高靖雅

2019/03/28

程式摘要 ：

此程式用 C++語言，

將字串與字串的 reverse 做比對，用 LCS 建立 DP 表格。

程式本體 & 註解 ：

```
#include <iostream>
#include <string>
#include <stdlib.h>
using namespace std;

char input[1000];      //輸入字串
int table[1000][1000] = { { -1 } }; //DP表格紀錄結果
int mark[1000][1000] = { {-1 } }; //DP表格紀錄結果

int main() {

	int n; //輸入有幾筆測資
	cin >> n;

	for (int i = 0; i < n; i++) {
		cin >> input;
		int len = strlen(input);//字串長度

		//table最上列及最左行初始化為0
		for (int j = 0; j <= len; j++)
			table[j][0] = 0;
		for (int j = 0; j <= len; j++)
			table[0][j] = 0;
```

```
            //做字串與字串reverse比對，以LCS填入table
            for (int j = 1; j <= len; j++) {
                for (int k = 1; k <= len; k++) {
                    if (input[j - 1] == input[len - k]) {
                        table[j][k] = table[j - 1][k - 1] + 1;
                        mark[j][k] = 0;  //若字元相同，標記為0
                    }
                    else {
                        if (table[j - 1][k] >= table[j][k - 1]) {
                            table[j][k] = table[j - 1][k];
                            mark[j][k] = 1;
                        }
                        else {
                            table[j][k] = table[j][k - 1];
                            mark[j][k] = 2;
                        }
                    }
                }
            }
            cout << table[len][len] << endl; //印出最大長度
            int j = len, k = len;
            while (1) {     // 依據LCS的table輸出結果
                if (j <= 0 || k <= 0)break;
                if (mark[j][k] == 0) {          //印出所求字串
                    cout << input[j - 1];
                    j--;
                    k--;
                }
                else if (mark[j][k] == 1)
                    j--;
                else if (mark[j][k] == 2)
                    k--;
            }
            cout << endl;
        }
        return 0;
}
```

Pseudo code:

```
LCS_Longest palindrome subsequence (X)
1    n = X.length
2    let a[1...n, 1...n] and b[1...n, 1...n] be new tables
3    for i = 1 to n
4         b[ i, 0 ] = 0
5    for j = 1 to n
6         b[ 0, j ] = 0
7    for i = 1 to n
8         for j =1 to n
9              if Xi == Yj
10                  b[ i, j ] = b[ i-1, j-1 ] + 1
11                  a[ i, j ] = "↖"
12             elseif b[ i-1, j ]>=b[ i, j-1 ]
13                  b[ i, j ] = b[ i-1, j ]
14                  a[ i, j ] = "↑"
15             else
16                  b[ i, j ] = c[ i, j-1 ]
17                  a[ i, j ] = "←"
18   return a and b
```

圖解　：

X 為輸入字串，Y 為輸入字串 reverse，逐一比對，若字相同，則該格子反藍

	j	0	1	2	3	4	5	6
i		Yj	B	D	C	A	B	A
0	Xi	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0	↖1	←1	↖1
2	B	0	↖1	←1	←1	←1	↖2	←2
3	A	0	↑1	↑1	↑1	↖2	↑2	↖3
4	C	0	↑1	↑1	↖2	↑2	↑2	↑3
5	D	0	↑1	↖2	↑2	↑2	↑2	↑3
6	B	0	↖1	↑2	↑2	↑2	↖3	↑3