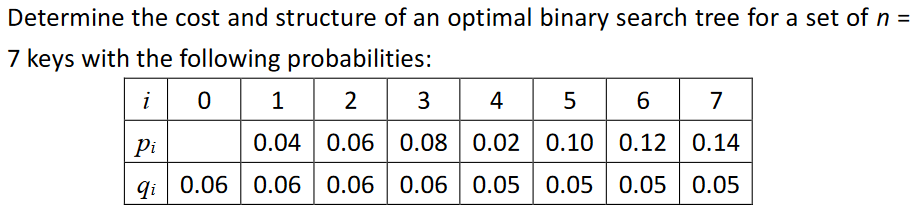
Algorithm

Hw5

Q1



w[i,j]=+

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| w | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0.16 | 0.28 | 0.42 | 0.49 | 0.64 | 0.81 | 1 |
| 2 |  | 0.18 | 0.32 | 0.39 | 0.54 | 0.71 | 0.9 |
| 3 |  |  | 0.2 | 0.27 | 0.42 | 0.59 | 0.78 |
| 4 |  |  |  | 0.13 | 0.28 | 0.45 | 0.64 |
| 5 |  |  |  |  | 0.2 | 0.37 | 0.56 |
| 6 |  |  |  |  |  | 0.22 | 0.41 |
| 7 |  |  |  |  |  |  | 0.24 |

E[i,j]=MINi<=r<=j{E[i,r-1]+E[r+1,j]+w[i,j]}

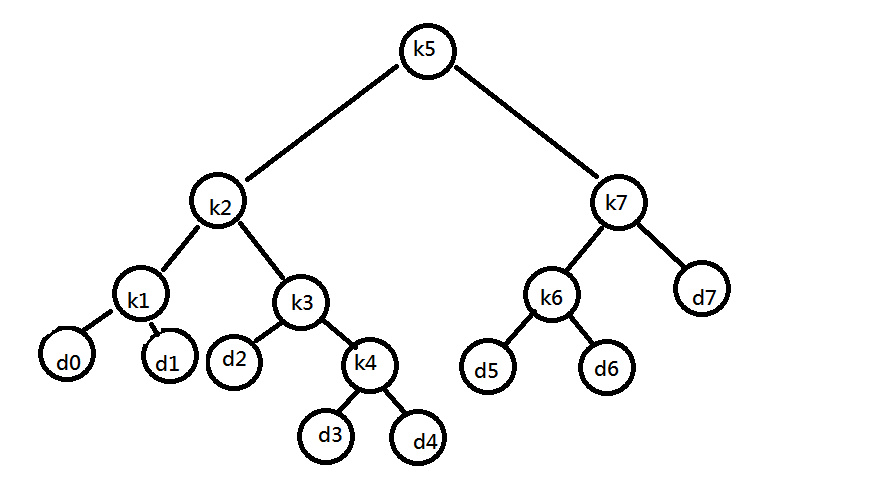
=q(i-1),if j=i-1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0.06 | 0.28 | 0.62 | 1.02 | 1.34 | 1.83 | 2.44 | 3.12 |
| 2 |  | 0.06 | 0.3 | 0.68 | 0.93 | 1.41 | 1.96 | 2.61 |
| 3 |  |  | 0.06 | 0.32 | 0.57 | 1.04 | 1.48 | 2.13 |
| 4 |  |  |  | 0.06 | 0.24 | 0.57 | 1.01 | 1.55 |
| 5 |  |  |  |  | 0.05 | 0.3 | 0.72 | 1.2 |
| 6 |  |  |  |  |  | 0.05 | 0.32 | 0.78 |
| 7 |  |  |  |  |  |  | 0.05 | 0.34 |
| 8 |  |  |  |  |  |  |  | 0.05 |

R[i,j]=a value of r that gives the MIN

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| R | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 2 | 2 | 2 | 3 | 3 | 5 |
| 2 |  | 2 | 3 | 3 | 3 | 5 | 5 |
| 3 |  |  | 3 | 3 | 4 | 5 | 5 |
| 4 |  |  |  | 4 | 5 | 5 | 6 |
| 5 |  |  |  |  | 5 | 6 | 6 |
| 6 |  |  |  |  |  | 6 | 7 |
| 7 |  |  |  |  |  |  | 7 |

Ans: cost=3.12

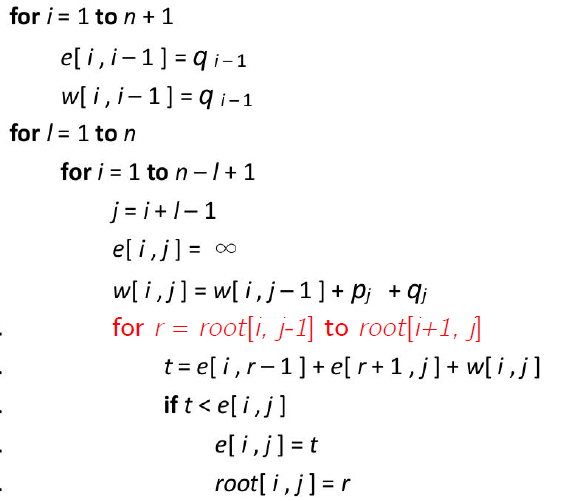


* Q2

遞迴式：e[i,j]=mini<=r<=j{e[i,r-1]+e[r+1,j] , w[i,j]}

Let e[1,…,n+1,0,…..n] , w[1,…,n+1,0,…..n] , root[1,….,n,1,…..,n] be the new table

Solution：



return e and root

Time complexity：最外層迴圈跑n次，內層迴圈跑次 , j=i+l+1

展開消去得r[n-l+2,n]-r[1,l-1]+n-l+1=O(n)

因此時間式n\*n=O(n^2)

* Q3

Recurrence relation:

L(i,j) = 0 , if i=j+1

= 1 , if i=j

= L(i+1,j-1)+2 , if i<j and s[i]=s[j]

=max{ L(i+1,j) , L(i,j-1) } , otherwise

Pseudocode:

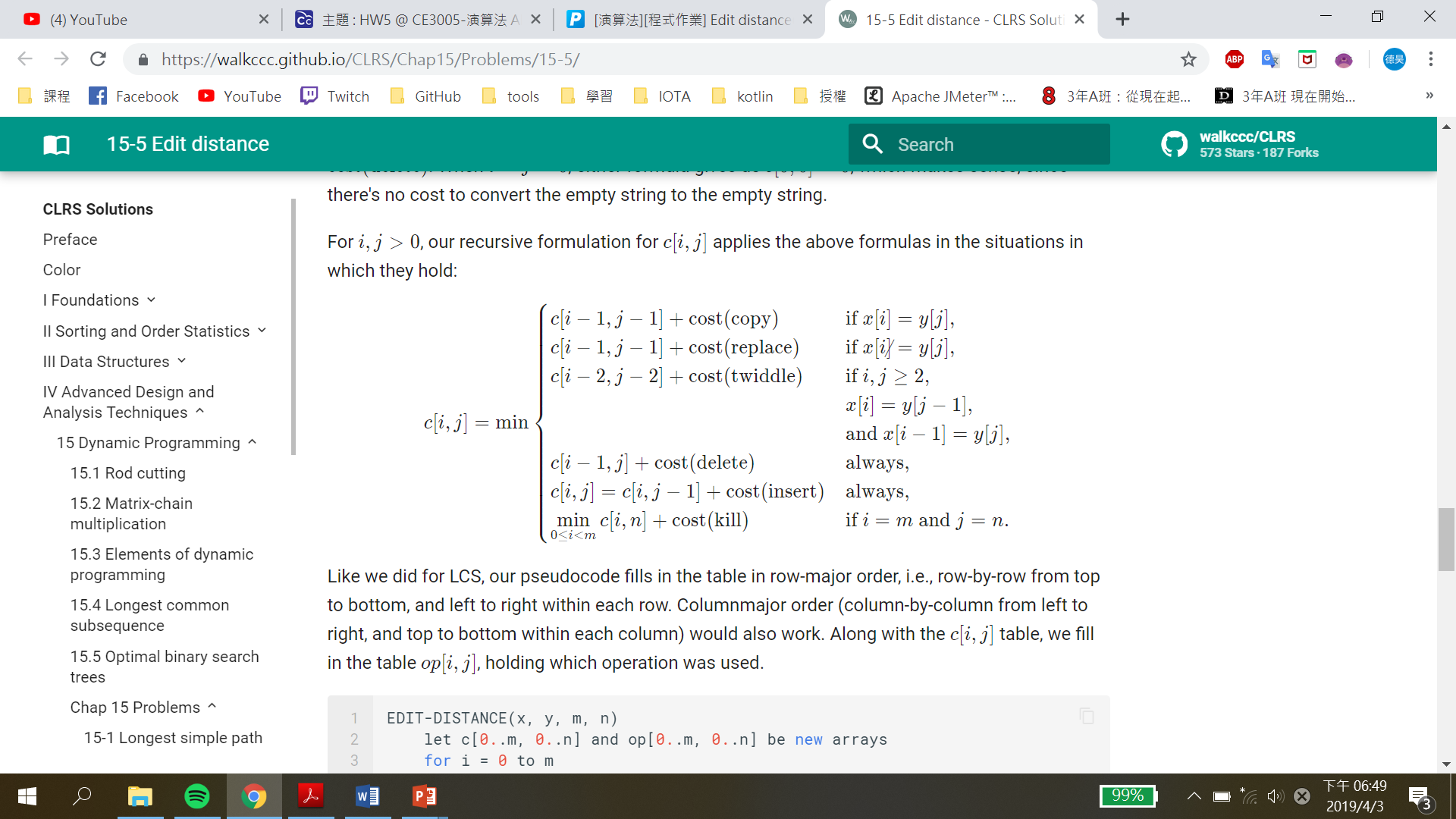
LPS(s)

1. for i=1 to n
2. for j=1 to n
3. if i==j+1
4. L(i,j)=0
5. else if i==j
6. L(i,j)=1
7. else if i<j and s(i)==s(j)
8. L(i,j)=L(i+1,j-1)+2 and r(i,j)=‘🡽’
9. else if L(i+1,j) L(i,j-1)
10. L(i,j)=L(i+1,j) and r(i,j)=‘🡹’
11. else L(i,j)=L(i,j-1) and r(i,j)=‘🡺’

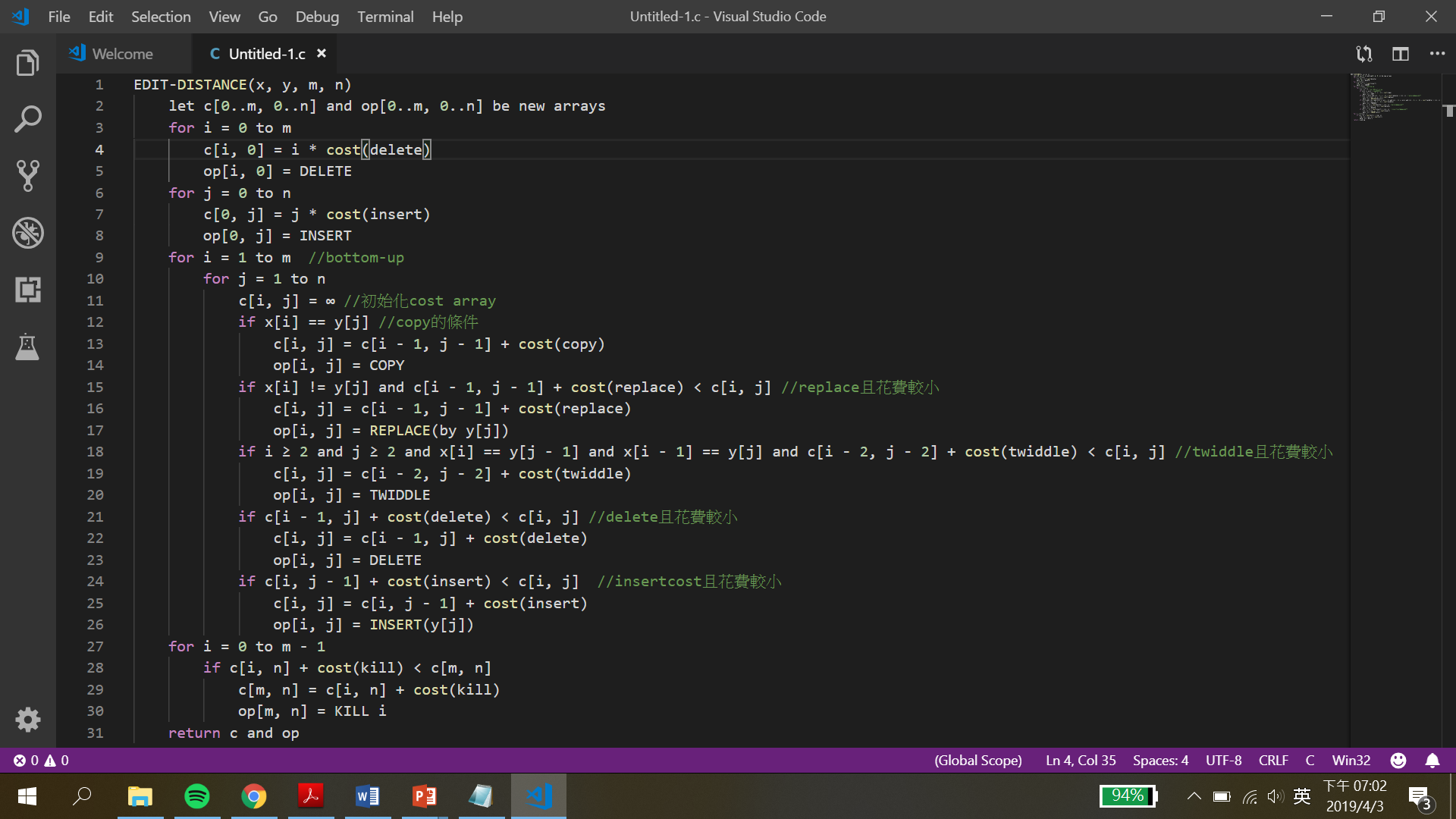
Time complexity: T(n)=

* Q4:

根據題目條件可推出以下遞迴式

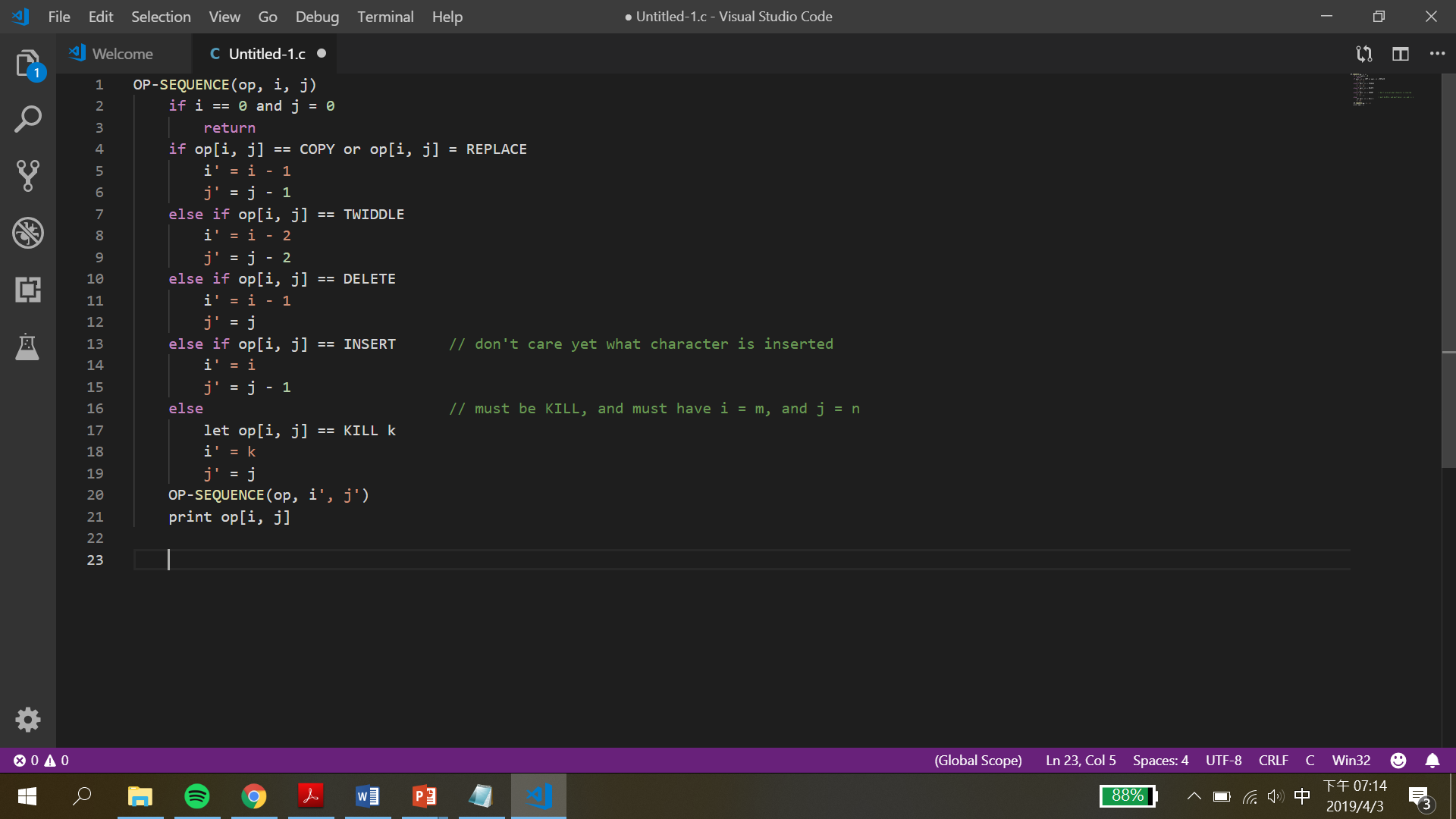


Pseudo Code  
0 ≤ i ≤m , 0 ≤ j ≤ n , m, n分別為兩字串長度

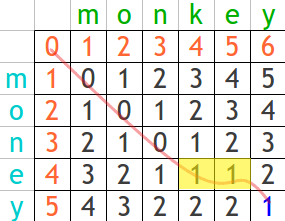


程式跑完後會將operation 和cost 分別記錄在op array和cost array，而在cost array[m][n]將為兩字串edit的最小cost。

Pseudo Code  
根據 EDIT-DISTANCE 得到的 op array 可印出 operation sequence



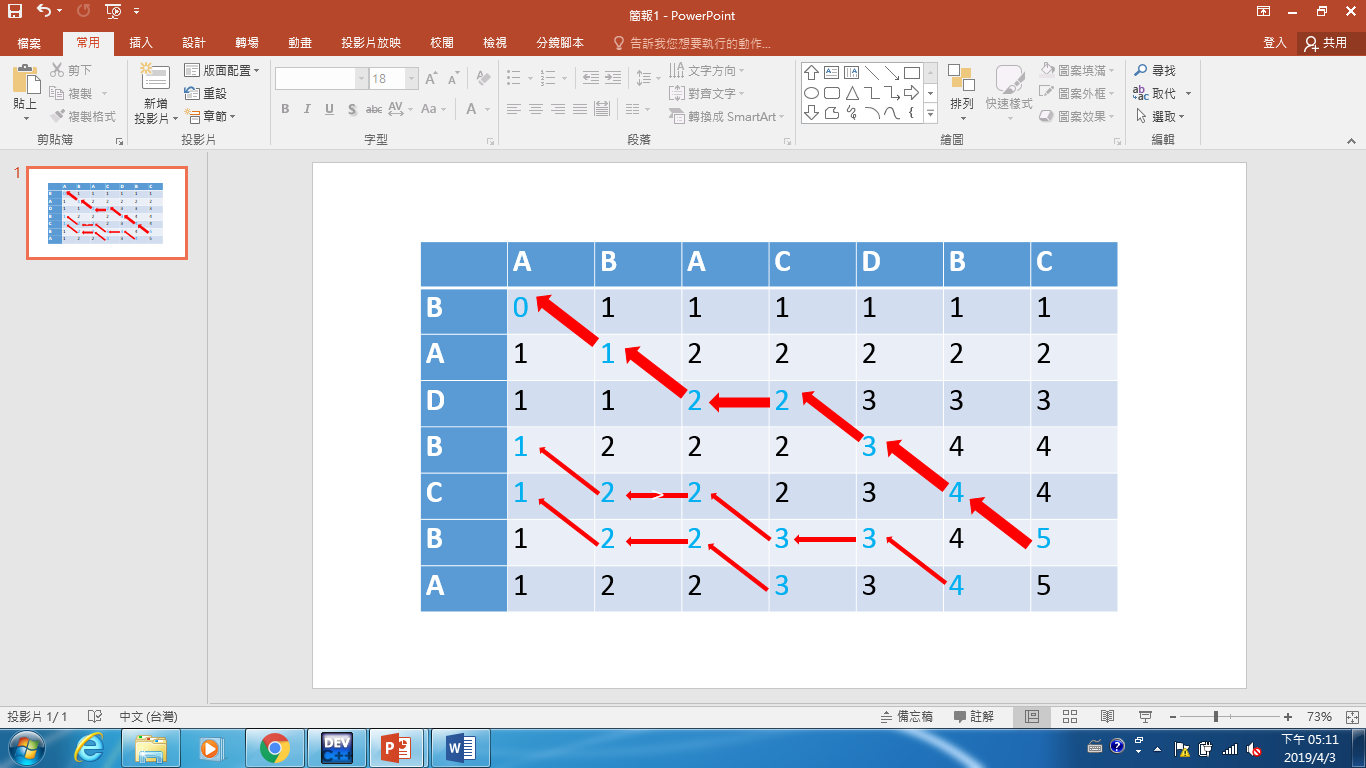
以下方例子舉例，紅線路徑為最小 cost  
operation sequence = copy-copy-copy-replace-delete(黃色部分)-copy  
cost = 4\*cost(copy)+cost(replace)+copy(delete)



時間複雜度 = O(m\*n) , m, n分別為兩字串長度

* Q5

Find out all LCS of《BADBCBA》and《ABACDBC》

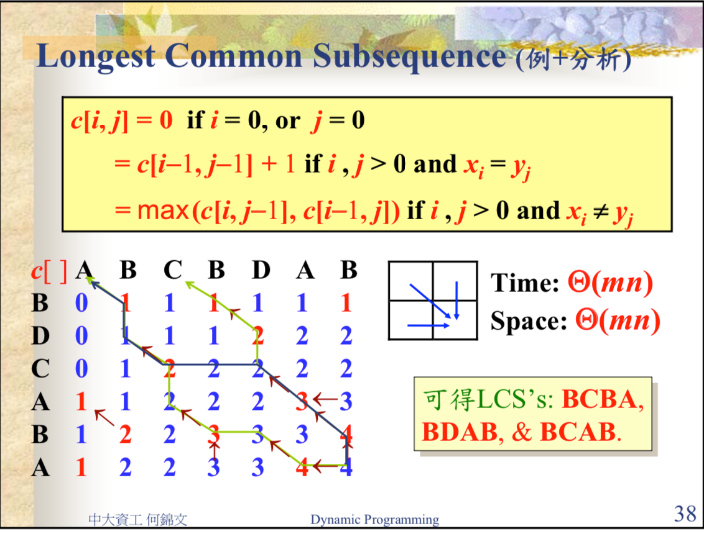


Ans: BADBC

* Q6

An algorithm to solve the LCS problem of two strings X and Y has space complexity O(|X||Y|) typically.

a. Design an algorithm to find the length of LCS of two string X and Y just using only 2\*min(|X|, |Y|) cells for working space.



回顧上課時尋找LCS的作法，經觀察後發現，每個元素得值只與自己的左邊與自己上面與自己的左上有關係，設計的構想為利用兩個矩陣a[]與b[]，a[]用來儲存前一row的元素，b[]用來儲存正在填的row的數值，在填b[i]時只需要看a[i-1],a[i],b[i-1]，遞迴的關係式為

b[i]

=a[i-1]+1 if x[i]=y[i]

=max(a[i],b[i-1]) if x[i]!=y[i]

以上面的例子而言每個步驟的值為

1

a={0,1,1,1,1,1,1} b={0,1,1,1,2,2,2}

2

a={0,1,1,1,2,2,2} b={0,1,2,2,2,2,2}

3

a={0,1,2,2,2,2,2} b={1,1,2,2,2,3,3}

4

a={1,1,2,2,2,3,3} b={1,2,2,3,3,3,4}

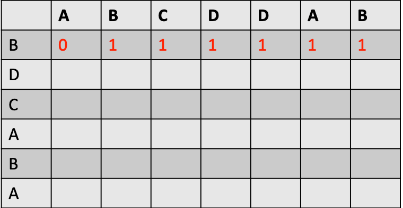
5

a={1,2,2,3,3,3,4} b={1,2,2,3,3,4,4}

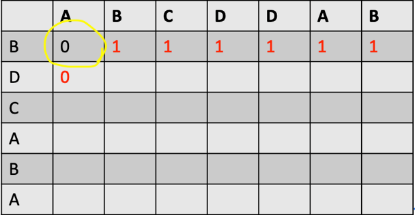
最後b陣列的最後一個元素就是答案。

b. Design an algorithm to find the length of LCS of two string X and Y just using only 1 + min(|X|, |Y|) cells for working space.

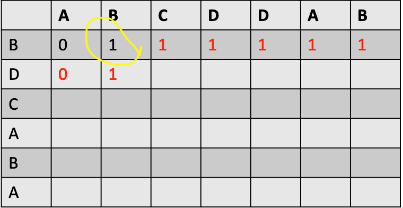
觀察a小題的作法，可以發現有一些a陣列中有一些元素在填b被看完時就用不到了，要等到b填完易整排時才會再被利用一次，因此我們可以把上一排準備要被看的那一個元素利用一個變數temp存起來，而剛剛被看完的一排。



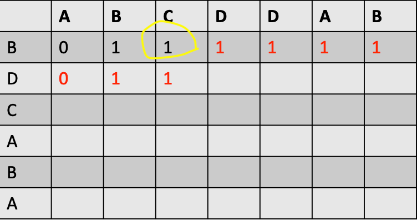
此時b={0,1,1,1,1,1,1} temp=0



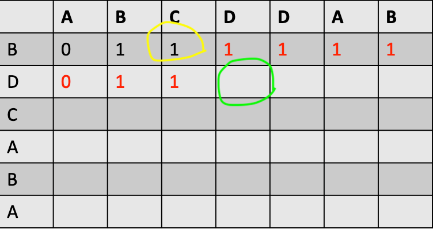
此時b={0,1,1,1,1,1,1} temp=0



此時b={0,1,1,1,1,1,1} temp=1



此時b={0,1,1,1,1,1,1} temp=1



在填綠圈這格時只要將temp+1就可，因為D==D

遞迴的關係是如下

b[i]

=0 if i=0

=temp+1 if x[i]=y[i] //temp 為b[i]左上方的元素

=max(b[i],b[i-1]) if x[i]!=y[i]

* Q7

F[I,j]=F[i-1,j]-1 if x[i]

=F[i-1,j-1]+1 if x[i]=y[j]

=F[i-1,j-1]-1 if y[i]y[j]

=F[I,j-1]-1 if y[j]