In normal implementation, $f(m, n)$ can find one of optimal alignment. How to find out all of optimal alignments ? :

Week 7
410821316
鄭禎文

For finding all possible optimal alignments, we can use the approach of Needlman-Wunsch Algorithm :

$$s1 = sub1 + \boxed{e1} \quad , \quad s2 = sub2 + \boxed{e2}$$

$$\left.\begin{array}{l} LCS(s1, s2) = LCS(s1, sub2) \\ LCS(s1, s2) = LCS(sub1, s2) \\ LCS(S1, S2) = LCS(sub1, sub2) \\ LCS(S1, S2) = LCS(sub1, sub2) + e1 \end{array}\right\} \text{4 condition}$$

$$\underbrace{\begin{array}{l} LCS(S1, S2) = \\ \quad \max(LCS(sub1, s2), LCS(s1, sub2), LCS(sub1, sub2)), \\ \qquad\qquad\qquad\qquad\qquad \text{when } e1 \neq e2 \\ \quad LCS(sub1, sub2) + e1, \quad \text{when } e1 = e2 \end{array}}_{\text{Recursive Function}}$$

$$LCS(s1, s2) = \phi, \quad \text{when } s1 \neq \phi \text{ or } s2 \neq \phi \; \} \text{ initial}$$

You can form a list using the algorithm. Within the forming progress, LCS might come from top, left, or top-left, store all the possible situation. In the tracing pack, traverse all of the possibilities and get all of the LCS, or optimal alignments. ※

How many optimal alignment may exist ? Please construct a set of input to explain your answer :

Input : ABCBDAB , BDCABA

Output : BCAB , BCBA , BDAB

		B	D	C	A	B	A
	0	0	0	0	0	0	0
A	0	↑0	↑0	↑0	↖1	←1	↖1
B	0	↖1	←1	←1	↑1	↖2	←2
C	0	↑1	↑1	↖2	←2	↑2	↑2
B	0	↖1	↑1	↑2	↑2	↖3	←3
D	0	↑1	↖2	↑2	↑2	↑3	↑3
A	0	↑1	↑2	↑2	↖3	↑3	↖4
B	0	↖1	↑2	↑2	↑3	↖4	↑4

BCBA

BCAB BDAB

⇒ Three optimal alignments exist . #

Suppose both A and B are very long, that we can't maintain all m×n scores in memory. Please find the way which only caches n values :

Two - dimensional array length [i][j], means first i elements of s1 and first j elements of s2.

In order to reduce the cost of memory space, we have to improve the way we create the list. For forming a list, we only need the upper block, lefter block and the left - upper block. For calculation, we set to be from left to right, then from top to bottom. By doing so, we only need one array (upper row) and a single variable (left - upper block), and its space complexity can be improved to O(min (N, M)), N and M to be the length of the sequence ※

Analyze space complexity , time complexity in best case and worst case in Q1 and Q_2 :

Time Complexity for Q1 :

⇒ $O(N \times M)$

N and M to be the sequence length *

Space Complexity for Q1 :

⇒ $O(N \times M)$

N and M to be the sequence length *

Time Complexity for Q2 :

⇒ $O(N \times M)$

N and M to be the sequence length *

Space Complexity for Q2 :

⇒ $O(N \times M)$

N and M to be the sequence length *