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

Week 7
41082316
詹穆文

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

$s1 = sub1 + (e1)$, $s2 = sub2 + (e2)$

$LCS(s1, s2) = LCS(s1, sub2)$
$LCS(s1, s2) = LCS(sub1, s2)$
$LCS(s1, s2) = LCS(sub1, sub2)$
$LCS(s1, s2) = LCS(sub1, sub2) + e1$

} 4 condition

$LCS(S1, S2) =$

$\max(LCS(sub1, s2), LCS(s1, sub2), LCS(sub1, sub2))$, when $e1 \neq e2$

$LCS(sub1, sub2) + e1$, when $e1 = e2$

Recursive Function

$LCS(s1, s2) = \phi$, when $s1 \neq \phi$ or $s2 \neq \phi$ } initial

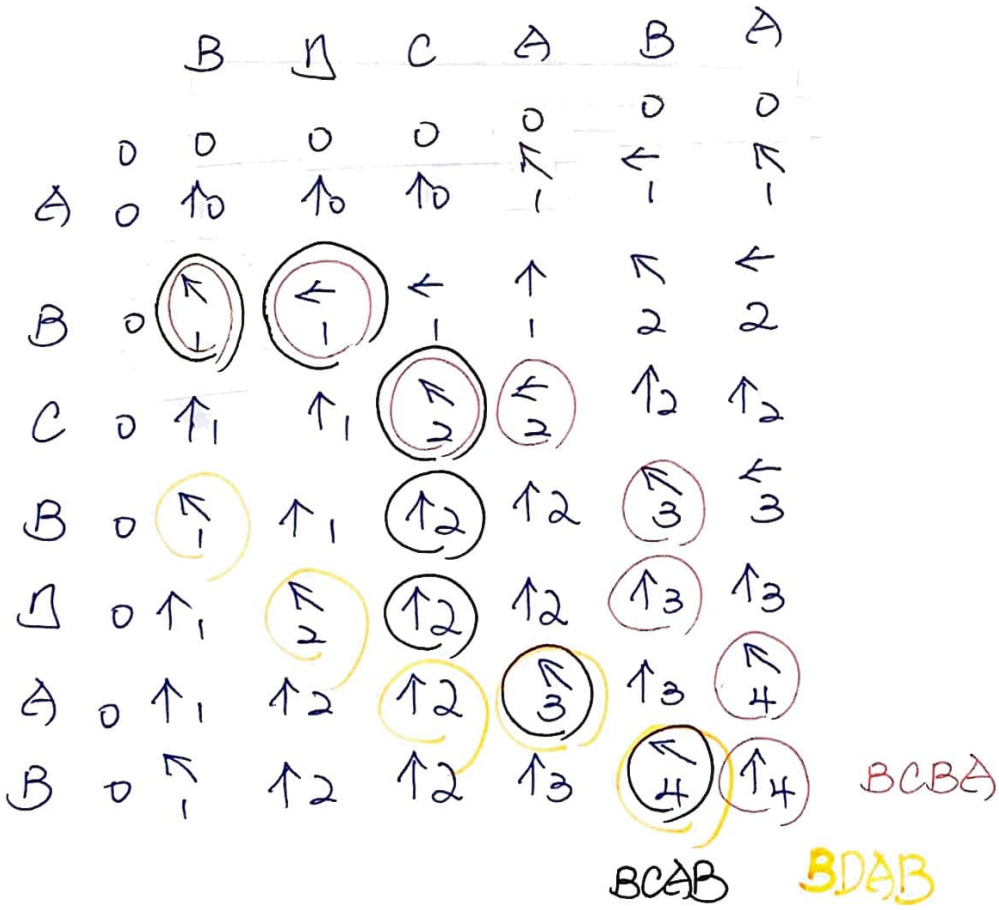
---

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



⇒ 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 *