Algorithm Lab

Week 7: Sequence Alignment

We have 2 sequences and . We can insert void symbols into either sequence at arbitrary place. An alignment is to make 2 sequences with the same length by do previous operation on and ; Assume the length of made sequences is , we can denote the sequence from by and denote the sequence from by . When we align and , for each pair symbol have 3 different states: **match** if ; **gap** if each of them is a void symbol; **mismatch** if . Every state will correspond to a score (or penalty), named , , and . Score of the alignment is the summation of score of each paired symbol. Sequence alignment problem ask to find an alignment can maximize the alignment score. Longest common subsequence problem is a specialized version of sequence alignment problem that , , and .

Instance: sequences and score

Result: sequences where and maximize the alignment score

# Description

We denote the substring of that contains leading symbols by and substring of that contains leading symbols by . For convenient, we define 3 functions as followed:

* : Find the alignment that can maximize alignment score of and .
* : The alignment score of .
* : Th alignment score of and . Should be or .

Since every symbol of and will be the trailing symbol of some or , thus, we can consider only trailing symbols. For . Under the policy, the only operation we need to try is append(insert) void symbol to , append void symbol to , or leave the original paring. If maps to a void symbol, means we’ll append a void symbol to ; if maps to a void symbol, we’ll append a void symbol to ; If we append nothing to and , then and should be paired and be match or be mismatch.

If nothing to match, our score should be 0, thus, and .

When implement this algorithm, we usually cache values of by a 2-dimension array.

# Questions

1. In normally implementation, can find one of optimal alignment. How to find out all of optimal alignments?
2. How many optimal alignments may exist? Please construct a set of input to explain your answer.
3. Suppose both and are very long, that we can’t maintain all scores in memory. Please find the way which only cache values.
4. Analyze space complexity, time complexity in best case and worst case in Q1 and Q2.
5. Solve <http://oj.csie.ndhu.edu.tw/problem/ALG04C>.