ASSIGNMENT - 02
COMP 544 - ALGORITHM
ALGORITHM ASSIGNMENT GROUP - 03

Dynamic Alignment Assignment:

1. Sub-Problems
   a. Break the problem into smaller subproblems.
   b. Solve the smaller problems optimally.
   c. Use the sub-problem solutions to construct an optimal solution for the original problem.

   For Example, let's assume that a = “abc” and b = “def.” For finding the minimum alignment for a and b[0:2], we take the best alignment for a and b[0:1] and use that information to find the best alignment for a and b[0:2]

2. Recurrence Relation

   Recurrence relation can be formed as follows:

   a. If i = 0 or j = 0;

   $$opt(i, j) = \delta * i$$
   $$opt(i, j) = \delta * j$$

   b. If i > 0 and j > 0;

$$opt(i, j) = min(a_{xi,yi} + opt(i - 1, j - 1), \delta + opt(i - 1, j), \delta + opt(i , j - 1))$$

3. Pseudocode

```
d ← Gap penalty score
for i = 0 to length(A)
    opt(i,0) ← d * i
for j = 0 to length(B)
    opt(0,j) ← d * j
```

```
for i = 1 to length(A)
    for j = 1 to length(B)
    {
        Match ← opt(i-1, j-1) + S(Ai, Bj)
        Delete ← opt(i-1, j) + d
        Insert ← opt(i, j-1) + d
        opt(i,j) ← min(Match, Insert, Delete)
    }
```

4. Example

We will walk through the example of a = “abc” and b = “dea”
Let’s assume that

I. The cost for Mismatch is 1
II. The cost for the GAP Penalty is 2
III. The cost for Match is 0.

At index i = 0 and j = 0, we have found a mismatch (a, d), so here we have 2 options: take the gap penalty or mismatch penalty. We will take a mismatch penalty since it has a minimum penalty.

At index i < j, we are trying to match a[0:i] with b[0:j], since length of b[0:j] is greater than a[0:i] we might take the minimum penalty from GAP penalty or mismatch penalty

At index i = 2 and j = 2, we either

I. match a from a[0] and take the 2 gap penalties for b[0] and b[1], which costs 4
II. Don't match with a[0] and take a mismatch penalty for b[0], b[1], and b[2], which costs 3
III. As we aim for the minimum sequence alignment cost, we choose the 2nd option and end up with a total cost of 3.

5. Python Code

https://drive.google.com/file/d/1vJC96iPQRMQW6q_4_faVgwzeQZ2qlhDy/view?usp=sharing