Hw #6 - due 11/13/Sun midnight

LCS → 1. <BDCABA>
opt sub property <ABCBDAB>
↑
bottom-up

Question 1.

we want to find a longest common sequence from these two. by using a table

How to use table?

Ex of Solving a problem by structural property.

Question 1. Answers

(1) allocate m+1 by n+1 matrix, where m and n are the lengths of the sequences.
(2) all entries of the first row (for index 0), and all entries of the first column (for index 0) are filled with 0's.
(3) starting from the second row, (for index 1) fill in each entry in the row using the equation (15.9) of the textbook on page 395,

which is

$$c[i,j] = \begin{cases} 0 & i = 0 \text{ or } j = 0 \\ c[i-1, j-1] + 1 & i, j > 0 \text{ and } x_i = y_j \\ \max(c[i,j-1], c[i-1,j]) & i, j > 0 \text{ and } x_i \neq y_j \end{cases}$$

which each entry is determined along the way, place a one of the following "marks" in the corresponding entry:

- if $x_i = y_j$ mark with "↖"
  if $c[i,j]$ is greater than or equal to $c[i,j-1]$, mark with "↑"
  else mark with "←"

(4) At the end of the construction, trace the "marks" starting from $c[m+1, n+1]$.

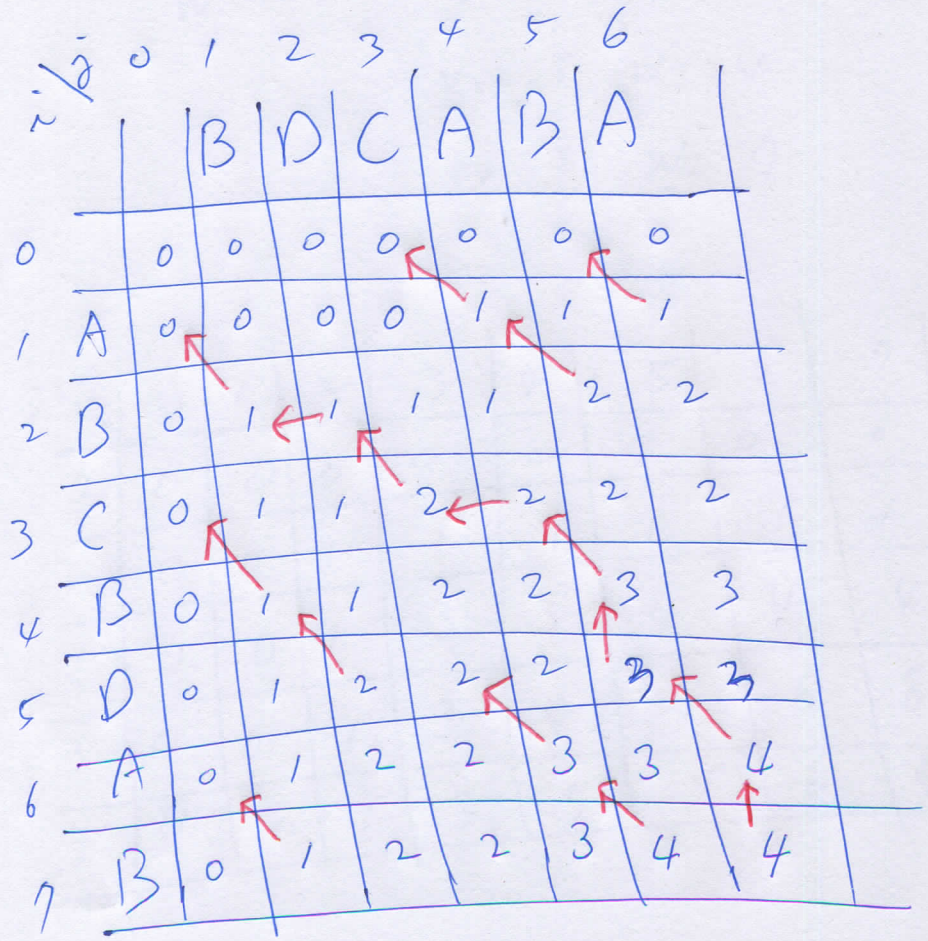
there are 3 possible marks recorded

a) # → two letters are identical

b) "A" → we need to look at the one right above the current entry

c) "←" → we need to look at the one right next to the current entry (to the left)

table C



order

1. construction of the table c — fill in 1st row, 2nd row, ⋮

2. trace the "marks" from c[m, n]
   m, n — in this example

Question2

0/1 knapsack problem solve by Dynamic programing by using a table.
n, v, w

		$	lb
6	item 1	60	10
5	item 2	100	20
4	item 3	120	30
			60

DP $/lb criteria 로 사용하여 item 1, 2 만 선택하여 item 3 을 사용하지 못한다. (GA X) 이렇게 table을 사용하여 풀 수 있을까?

Question 2 Answers

optimal value of the 0/1 knapsack problem.

Given n items and the weight for the knapsack w, allocate a table that consists of n+1 rows and w+1 columns.

let's call this table v.

All entries on the first row & columns are filled with 0's

use the following equations to compute the value

$$v(i,w) = \max(v(i-1,w), v_i + v(i-1, w-w_i)) \quad \text{if } w_i \le w$$

(이전값 or)

$v(i-1,w)$ if $w_1 > w$

The optimal value will be in the entry $v[n+1, w+1]$.

$v_1 = 1 \quad w_1 = 2$

$W = 10 \qquad v_2 = 3 \quad w_2 = 3$

$v_3 = 5 \quad w_3 = 4$

$v_4 = 9 \quad w_4 = 7$