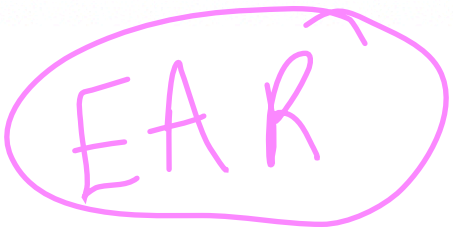
- Compute the Longest Common String (LCS) between **LEAR and EARN** using **dynamic programming (DP) bottom-up approach. (40 pts)**
  - Fill in the matrix and compute the length of the LCS. Include backtracking arrows or mark
  - Backtrack to determine the LCS

	j	0	1	2	3	4
i	0	0	E	A	R	N
0	0	0	0	0	0	0
1	L	0	0	0	0	0
2	E	0	1	1	1	1
3	A	0	1	2	2	2
4	R	0	1	2	3	3



- Compute the Longest Common String (LCS) between **ALGORITHM and ALIGNMENT** using **dynamic programming (DP) bottom-up approach. (60 pts)**
  - Fill in the matrix and compute the length of the LCS. Include backtracking arrows or marks
  - Backtrack to determine the LCS

← Cross the diagonal ←: I can go up or down

	j	0	1	2	3	4	5	6	7	8	9
i			A	L	I	G	N	M	E	N	T
0		0	0	0	0	0	0	0	0	0	0
1	A	0	1	← 1	← 1	← 1	← 1	← 1	← 1	← 1	← 1
2	L	0	↑ 1	2	← 2	← 2	← 2	← 2	← 2	← 2	← 2
3	G	0	↑ 1	↑ 2	← 2	3	← 3	← 3	← 3	← 3	← 3
4	O	0	↑ 1	↑ 2	← 2	↑ 3	← 3	← 3	← 3	← 3	← 3
5	R	0	↑ 1	↑ 2	← 2	↑ 3	← 3	← 3	↑ 3	↑ 3	← 3
6	I	0	↑ 1	↑ 2	3	← 3	↑ 3	↑ 3	← 3	← 3	↑ 3
7	T	0	↑ 1	↑ 2	↑ 3	← 3	↑ 3	← 3	↑ 3	← 3	4
8	H	0	↑ 1	↑ 2	↑ 3	← 3	← 3	← 3	← 3	← 3	↑ 4
9	M	0	↑ 1	↑ 2	↑ 3	↑ 3	↑ 3	4	← 4	← 4	← 4

(ALIM)