

## CS 361 – Lab #10 – Dynamic Programming

In today's lab, you will implement the longest-common-sequence algorithm we discussed in class. Here is the pseudocode:

```
m = length(X)
n = length(Y)
for i = 0 to m:
    len[i, 0] = 0
for j = 0 to n:
    len[0, j] = 0
for i = 1 to m:
    for j = 1 to n:
```

If/else condition	Set len[i,j] to	Set dir[i,j] to
x[i] == y[j]	1 + len[i-1,j-1]	NW
len[i-1,j] >= len[i,j-1]	len[i-1,j]	N
Otherwise	len[i,j-1]	W

Create a new directory called Lab10. Copy my source file LCS.java from my Lab10 folder on the Web. This is the only source file you need to modify.

The program is designed to read 2 input files specified by the user. The elements in the sequences and subsequences will be tokens representing individual words. Test your program with two similar text documents. Relative to the sizes of the input documents, how long of a subsequence would we expect if one document was essentially a copy of the other? Consider that a handful of words were changed, some phrases inserted and deleted.

Run your program with example input files I've provided. Does the output match what you would have expected?

---