CSE 5311-003 Lab Assignment 2

Due November 21, 2002

Goals:

1. Review of dynamic programming.

2. Understanding of a space-efficient implementation of longest common subsequences.

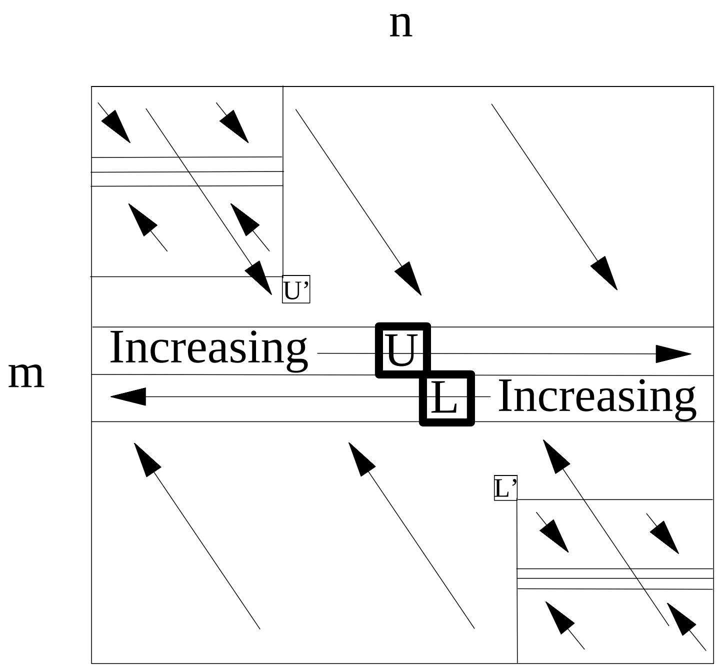
Requirements:

1. Implement a program to determine the longest common subsequence of two sequences of integers using O(m+n) space and O(mn) time. Your program must compile and execute on OMEGA.

2. Email your code (a single source file) as an attachment to `yxb4544@omega.uta.edu` before 2:00 pm on November 21. Your subject should include your name as recorded by the University.

Getting Started:

1. Your program should operate recursively and should not explicitly save back trace information. The following diagram demonstrates the concepts:



   a. Use the usual NW to SE cost computation to get increasing left-to-right cost row for LCS of the first m/2 elements of first sequence and entire second sequence.
   b. Perform symmetric (SE to NW) cost computation for LCS of the last m/2 elements of first sequence and entire second sequence. (This may be coded explicitly or may be handled indirectly by code to reverse arrays and reusing the NW to SE code.)
   c. Scan across the two seams for the diagonal pair (U,L) with the maximum sum. This sum is the length of the LCS.
   d. Scan left-to-right across the upper seam for the first entry U' with the U value. If non-zero, this gives an element to go into the LCS output array.
   e. Scan right-to-left across the lower seam for the last entry L' with the L value. If non-zero, this gives another element to go into the LCS output array.
   f. The U' and L' entries correspond directly to elements in the second sequence. Two simple scans can be used to find the corresponding elements of the first sequence.
   g. Call recursively for each of the two subproblems. One of the arguments will be the appropriate subarray of the LCS output array.

2. Your input (`stdin`) will consist of the following:

    a. A line with the values m and n. Neither value will exceed 5000, but still allocate storage dynamically.
    b. m lines with non-negative integers for the first sequence.
    c. -1
    d. n lines with non-negative integers for the second sequence.
    e. -1

3. Your output (`stdout`) will consist of the following:

    a. A line with the length of the LCS.
    b. The values in the LCS, one per line.
    c. -1