CSE331 homework 8 Solution (Greedy algorithm and dynamic programming)

1. (10 pts) We used dynamic programming to solve the longest common subsequence (LCS) problem in class. This algorithm needs to use a 2 dimensional array with size |x| * |y| given that x and y are two input sequences. Now, modify the dynamic programming algorithm so that we only need to use a linear table with size either |x| or |y|. Describe your idea (or modification) and then write your pseudocode.
Note: you don't need to find the longest common subsequence between x and y. You only need to find the size of an LCS.
Answer: when you fill in every cell of this matrix, it only uses pre-computed values in the previous row. Thus, you only need to keep two rows to get the final result.

2. (10 pts) Textbook (the 3rd version) problem 10.28. Please use the following pseudo-code to fill in the matrix M when s = 1 and 2.

```
//Compute minimum cost to multiply M_L, M_{L+1}, …, M_R
For  s =1 to R-L+1 //s is the number of input matrices
      for i = L to R+1-s
           for j = i+s-1
              if (i == j) M[i, j] = 0
                else
              M[i, j] = min i<=k<=j-1 { M[i,k] + M[k+1,j]+c_{i-1}c_k c_j }
```

Answer: 1150 scalar multiplications are used if the order of evaluation is
((A1A2)(((A3A4)A5)A6))

3. (5 pts) Textbook (the 3rd version) problem 10.29 (a). In order to show that this greedy choice does not work, provide a counter-example.
Answer: (a) Let the chain be a 1x1 matrix, a 1xA matrix, and an AxB matrix. Multiplication by using the first two matrices first makes the cost of the chain A + AB . The alternative method gives a cost of AB + B , so if A > B , then the algorithm fails. Thus, a counterexample is multiplying a 1x1 matrix by a 1x3 matrix by a 3x2 matrix.
Or, you can try to build a counter example of three matrices of sizes: A1: c0xc1, A2: c1xc2, and A3: c2xc3. And, you can assume that c1*c2*c3 < c0*c1*c2. But the optimal solution is (A1A2)A3 instead of A1(A2A3).

4. [10 pts] Textbook (the 3rd version) problem 10.35.
Answer: follow our example of Fibonacci numbers. Use a 2D matrix C of size N by k. The result of C(N,k) is the value in matrix C[N,k].

4. (25 pts) This is a programming problem. Implement the dynamic programming algorithm to find the longest common subsequence between two input sequences. You need to fill in the table and then do the trace-back in order to output the LCS. Below you can find the specific requirements:

1) The program should be named as LCS and take two files as inputs.

   LCS inputfile1 inputfile2

2) Both files contain a single-line string of numbers only, such as 10 22 33 etc.
3) You file should output the LCS between the two input strings.
4) Two example input files can be found at
   http://www.cse.msu.edu/~cse331/examples/inputfile1 (inputfile2)
5) We will test your program using two different files.