**CS430 Lecture 15 Activities**

Opening Questions

1. Why are optimal solutions to sub-problems stored in a table in dynamic programming solutions?

Constructing the answer for the Optimal Matrix Chain Multiplication (optimal parenthesization) from the dynamic programming table. See the solution to the example problem  
A1 A2 A3 A4 A5 A6

30x35 35x15 15x5 5x10 10x20 20x25

|  |  |
| --- | --- |
| P table | root table |

2. Write the optimal parenthesization.

3. Write pseudocode to use the root table to print the optimal parenthesization. Then write pseudocode to use the root table to actually perform the multiplications in the optimal parenthesization.

Longest Common Subsequence

Example: X[1…..m] Y[1…..n]

X : ABCBDAB Y : BDCABA

Length of LCS = 4 BCBA or BCAB

1. The brute force approach would be to find all subsequences of one input, see if each exist in other input. How many are there?

2. Step 1: Generically define the structure of the optimal solution to the Longest Common Subsequence problem.

The longest common subsequence of sequence X[1…..m] and sequence Y[1…..n] is:

3. Step 2: Recursively define the optimal solution. Assume C(i,j) is the optimal answer for up to position i in X and position j in Y. Make sure you include the base case.

4. Use proof by contradiction to show that Longest Common Subsequence problem has optimal substructure, i.e. the optimal answer to problem must contain optimal answers to sub-problems.

5. Step 3: Compute solution using a table bottom up for the Longest Common Subsequence problem. Use your answer to question 3 above. Note the overlapping sub-problems as you go. .Step 4: Construct Optimal solution

X = ABCBDAB Y = BDCABA

<https://www.cs.usfca.edu/~galles/visualization/DPLCS.html>