CSC373 Worksheet 3 Solution

July 28, 2020

1. **Notes:**

- Dynamic Programming
  - Is applied to optimization problems
  - Applies when the subproblems overlap
  - Uses the following sequence of steps
    1. Characterize the structure of an optimal solution
    2. Recursively define the value of an optimal solution
    3. Construct an optimal solution from computed information

- Matrix-chain Multiplication
  - Is an optimization problem solved using dynamic programming
  - Goal is to find matrix parenthesis with fewest number of operations

    Example:

    Given chain of matrices $< A, B, C >$, it's fully parenthesized product is:

    * $(AB)C$ needs $(10 \times 30 \times 5) + (10 \times 5 \times 60) = 1500 + 3000 = 4500$ operations
    * $A(BC)$ needs $(30 \times 5 \times 60) + (10 \times 30 \times 60) = 27000$ operations

    Thus, $(AB)C$ performs more efficiently than $A(BC)$.

  - Is stated as: given a chain $< A_1, A_2, ..., A_n >$ of n matrices, where for $i = 1, 2, ..., n$ matrix A_i has dimension $p_{i-1} \times p_i$, fully parenthesize the product $A_1 A_2 ... A_n$ in a way that minimizes the number of scalar multiplications.
  - Steps

1. Check is the problem has Optimal Substructure

   <u>Example (From CS Breakdown on Youtube):</u>

   Let us adopt the notation $A_{i...j}$ where $i \leq j$, for the matrix that results from evaluating the product $A_i A_{i+1} ... A_j$.

   Assume the solution has the following parentheses:

   $$(A_{i...k})(A_{k+1...j})$$

   If there is a better way to multiply $(A_{i...k})$, then we would have a more optimal solution.

   This would be a contradiction, as we already stated that we have the optimal solution for $A_{i...j}$.

   Therefore, this problem has optimal substructure.

2. Recursive Solution
3. Computing the Estimated Cost
4. Constructing the Optimal Solution