CS 430 Lecture 14 Activities

Opening Questions

1. Briefly explain what two properties a problem must have so that a dynamic programming algorithm[1][2] will work.

   - Optimal Substructure
     - Optimal solution to the problem contains optimal solutions to subproblems.
   - Overlapping Subproblems
     - Those optimal solutions to subproblems are possibly used more than once.

2. Previously we have learned that divide-and-conquer algorithms partition a problem into independent sub-problems, solve each sub-problem recursively, and then combine their solutions to solve the original problem. Briefly, how are dynamic programming algorithms similar and how are they different from divide-and-conquer algorithms? Dynamic programming algorithms are optimization problems only, where divide and conquer algorithms could be used for more than optimization problems like sorting. Another difference is that divide and conquer problems only need to solve the subproblems that the divide creates.

   Solving

   - What are all possible subproblems when sorting?
   - Mergesort creates 2 subproblems of length $\frac{n}{2}$.
   - Quicksort creates 2 subproblems where one partition holds items $\leq$ the pivot and the other partition contains $>$.
   - But you could also create a subproblem with 1 item on one side $n-1$ items in the other, or 2 items on one side and $n-2$ items on the other.

   For sorting problems, you only need to choose one way to divide the problem, but dynammic programming requires that you consider all the ways to divide the problem; you only keep track to the answer of the optimal one due to optimal substructure. Dynammic programming stores the optimal answers in a table to use later[3] while divide-and-conquer would recalculate the answer during each recursion.

3. Why does it matter how we parenthesize a chain of matrix multiplications? We get the right answer any way we associate the matrices for multiplication. i.e. If A, B and C are matrices of correct dimensions for multiplication, then $(A \times B)C = A(B \times C)$. It matters which order we parenthesize a chain of matrix multiplications since the runtime might be larger for different configurations. If A has the dimensions 3×4, B has the dimensions 4×7, and

[1] Used for optimization problems.
[2] Have an exponetial problem space/solution space.
[3] Due to the overlapping subproblems.

C has the dimensions 7×3, multiplying the chain as $(AB)C$ would require 147 operations[4], whereas $A(BC)$ would take 120 operations[5], 27 operations less than $(AB)C$.

Dynamic Programming

Dynamic Programming Steps

1. Define structure of optimal solution, including what are the largest sub-problems.
2. Recursively define optimal solution
3. Compute solution using table bottom up
4. Construct Optimal Solution

Optimal Matrix Chain Multiplication (optimal parenthesization) (Minimize the work in the matrix multiplication which depends on matrix dimensions.)

1. How many ways are there to parenthesize (two at a time multiplication) 4 matrices $A \times B \times C \times D$?

   (1) $ABCD$
   (2) $(AB)(CD)$
   (3) $A((BC)D)$
   (4) $(A(BC))D$

   The general form is the different permutations $(n-1)!$ of the order of multiplications, however, some of them are identical in the work to do the multiplications.

2. Step 1: Generically define the structure of the optimal solution to the Matrix Chain Multiplication problem. The optimal way to multiply n matrices A_1 through A_n is: Problem: $A_1 \times A_2 \times A_3 \times \cdots \times A_n$. Assumed optimal parenthesization (solution): $(A_1 \dots A_k)(A_{k+1} \dots A_n)$ where the final multiplication is between these two subproblems, $1 \leftarrow k \rightarrow n-1$.

3. Step 2: Recursively define the optimal solution. Assume $\mathrm{P}(1, n)$ is the optimal cost answer. Make sure you include the base case. $P(1,n) = \min_{1 \leftarrow k \rightarrow n-1} \{\mathrm{P}(1,k) + \mathrm{P}(k+1,n) + r_1 c_k c_n\}$, $P(i,i) = 0$.

4. Use proof by contradiction to show that Matrix Chain Multiplication problem has optimal substructure, i.e. the optimal answer to the problem must contain optimal answers to sub-problems. Problem $A_1 \dots A_n$: what is optimal parenthesization? Assume you have an optimal answer, as we've already stated, $(A_1 \dots A_k)(A_{k+1} \dots A_n)$ for some k between 1 and $n-1$. If we have some other parenthesization such that $(A_1 \dots A_m)(A_{m+1} \dots A_n)$ where $m \neq k$ that is a better parenthesization than $(A_1 \dots A_k)(A_{k+1} \dots A_n)$. This would contradict our assumption that we already have our most optimal solution at $(A_1 \dots A_k)(A_{k+1} \dots A_n)$.

[4] Multiplying a 3×4 matrix by a 4×7 matrix results in a 3×7 that would be calculated in $3 \times 7 \times 4 = 84$ operations. Then doing the $(3 \times 7)(7 \times 3)$ would take $3 \times 7 \times 3 = 63$ operations. The initial 84 operations + 63 operations totals to 147 operations.

[5] $(4 \times 7) \times (7 \times 3) \rightarrow 4 \times 7 \times 3 = 84$ operations and a resulting 4×3 matrix. $(3 \times 4)(4 \times 3) \rightarrow 3 \times 4 \times 3 = 36$ operations and a total of 120 operations.

5. Step 3: Compute solution using a table bottom up for the Matrix Chain Multiplication problem. Use your answer to question 3 above. Note the overlapping sub-problems as you go.

6. Step 4: Construct Optimal solution

A	B	C	D
2×4	4×6	6×3	3×7

Table 14.1: Optimal Binary Search Tree Table: Memory $O\left(\frac{n^2}{2}\right)$

P(row, col)		1 A	2 B	3 C	4 D
1	A	0	48[6] $k=1$	84[7] $k=2$	126[8] $k=3$
2	B		0	72[9] $k=2$	156[10] $k=3$
3	C			0	126[11] $k=3$
4	D				0

[6] $2 \times 4 \times 6 = 48$

[7]
$$A(BC)_{k=1} = 0 + 72 + (2 \times 4 \times 3) \quad = 72 + 24 = 96$$
$$(AB)C_{k=2} = 48 + 0 + (2 \times 6 \times 3) \quad = 48 + 36 = 84$$

[8]
$$A(BCD)_{k=1} = 0 + 156 + (2 \times 4 \times 7) \quad = 156 + 56 = 212$$
$$(AB)(CD)_{k=2} = 48 + 126 + (2 \times 6 \times 7) \quad = 48 + 126 + 84 = 258$$
$$(ABC)D_{k=3} = 84 + 0 + (2 \times 3 \times 7) \quad = 84 + 42 = 126$$

[9] $4 \times 6 \times 3 = 72$

[10]
$$B(CD)_{k=2} = 0 + 126 + (4 \times 6 \times 7) \quad = 126 + 168 = 294$$
$$(BC)D_{k=3} = 72 + 0 + (4 \times 3 \times 7) \quad = 72 + 84 = 156$$

[11] $6 \times 3 \times 7 = 126$