Dynamic Programming

- Optimization problems

  "Hallmarks":
  - Optimal substructure
  - Overlapping sub problems

- Longest common subsequence : LCS
  - LLCS := length of LCS

$$LLCS(X, Y, i, j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ LLCS(X, Y, i-1, j-1) & \text{if } X[i] = Y[j] \\ \max\{LLCS(X, Y, i-1, j), \\ \quad LLCS(X, Y, i, j-1)\} & \text{if } X[i] \neq Y[j] \end{cases}$$

Recursive alg. repeatedly solves the same problems

Matrix Multiplication

A $n \times m$-matrix, B $m \times k$-matrix

$\Rightarrow$ $C := A \cdot B$ is $n \times k$-matrix

Entry at position (i,j):

$$c_{ij} = \sum_{\ell=1}^{m} a_{i\ell} b_{\ell j}$$

Needs m scalar multiplications

Computing C takes $n \cdot m \cdot k$ multiplications

Bracketing of Matrix Products

$$A \cdot B \cdot C \cdot D = ((AB)C)D = (AB)(CD) = A(B(CD))$$

$$(A(BC))D = A((BC)D)$$

Cost depends on bracketing (parenthesization)

Ex: A 30×1, B 1×40, C 40×10, D 10×25

- $((AB)C)D = 30 \cdot 1 \cdot 40 + 30 \cdot 40 \cdot 10 + {}$
  $+ 30 \cdot 10 \cdot 25$
  $= 1200 + 12\,000 + 7500$

- $A((BC)D) = 1 \cdot 40 \cdot 10 + 1 \cdot 10 \cdot 25 + 30 \cdot 1 \cdot 25$
  $= 400 + 250 + 750$

Which bracketing is optimal?

Bracketing Problem

- matrices $A_1, \ldots, A_n$ s.th.
  - A_1 is $p_0 \times p_1$ - matrix
  - A_2 is $p_1 \times p_2$ - matrix
  - ...
  - A_n is $p_{n-1} \times p_n$ - matrix
- find bracketing that minimizes cost of multiplications

Number of Bracketings

$C(n) :=$ # of bracketings for $A_1 \ldots A_n$

$$C(n) = \begin{cases} 1 & n = 1 \\ \sum_{k=1}^{n-1} C(k)\, C(n-k) & n > 1 \end{cases}$$

Recursion defines Catalan numbers

$$C(n) = \frac{1}{n+1}\binom{2n}{n} = \prod_{k=2}^{n} \frac{n+k}{k}$$

$$= \Omega\left(\frac{4^n}{n^{3/2}}\right)$$

Also

- number of binary trees with n nodes

Brute force is impossible!

Observation

Suppose optimal bracketing splits

$$(*) \qquad (A_1 \cdots A_k)(A_{k+1} \cdots A_n)$$

Then the bracketing for $A_1 \cdots A_k$ in $(*)$ is optimal for $A_1 \cdots A_k$ (similarly for $A_{k+1} \cdots A_n$)

Hallmark #1

Definition

$M(i,j) :=$ minimum number of mults necessary to compute $A_i \cdots A_j$

$\Rightarrow$ uses <u>some</u> bracketing $(A_i \cdots A_k)(A_{k+1} \cdots A_j)$

but we don't know k

Recursion

$$M(i,j) = \begin{cases} 0 & \text{if } i=j \\ \min\limits_{i \le k < j} \left\{ M(i,k) + M(k+1,j) + p_{i-1} p_k p_j \right\} & \text{if } i<j \end{cases}$$

Hallmark #2

Auxiliary function

$S(i,j) := k$, k optimal split for $A_i \cdots A_j$

<u>Subproblems</u>

- determined by (i,j), $1 \le i \le j \le n$

How many?

- $1 \le i < j \le n$: $\binom{n}{2}$
- $1 \le i = j \le n$: n

(6)

Example

$p[0,4] = [30, 1, 40, 10, 25] = p_0 \cdots p_4$

$M(i,j)$

	1	2	3	4
1	0	1,200	700	1,400
2	—	0	400	650
3	—	—	0	10,000
4	—	—	—	0

$$M(1,2) = M(1,1) + M(2,2) + p_0 p_1 p_2$$
$$= 0 + 0 + p_0 p_1 p_2$$

$$M(2,3) = p_1 p_2 p_3$$

$$M(3,4) = p_2 p_3 p_4$$

$$M(1,3) = \min \{ M(1,1) + M(2,3) + 30 \cdot 1 \cdot 10, \; M(1,2) + M(3,3) + 30 \cdot 40 \cdot 10 \}$$

$$= \min \{ 0 + 400 + 300, \; 1,200 + 0 + 1,200 \} = 700$$

7

$$M(2,4) = \min\{ M(2,2) + M(3,4) + p_1 \cdot p_2 \cdot p_4,$$
$$M(2,3) + M(4,4) + p_1 \cdot p_3 \cdot p_4 \}$$

$$= \min\{ 0 + 10{,}000 + 1 \cdot 40 \cdot 25,$$
$$400 + 0 + 1 \cdot 10 \cdot 25 \}$$

$$= 650$$

$$M(1,4) = \min\{ M(1,1) + M(2,4) + p_0 p_1 p_4,$$
$$M(1,2) + M(3,4) + p_0 p_2 p_4,$$
$$M(1,3) + M(4,4) + p_0 p_3 p_4 \}$$

$$= \min\{ 0 + 650 + 30 \cdot 1 \cdot 25,$$
$$1{,}200 + 10{,}000 + 30 \cdot 40 \cdot 25,$$
$$700 + 0 + 30 \cdot 10 \cdot 25 \}$$

$$= 650 + 750 = 1400$$

Example (cntd)

$s(i,j)$ tracks split points

	2	3	4
1	1	1	1
2	—	2	3
3	—	—	3