**Dynamic Programming**

**Matrix-Chain Multiplication**

Recall `Mat_mult(A, B) A \cdot B = C`

For `i = 1` to `A.rows`

For `j = 1` to `B.cols`

\[ C_{ij} = \sum_{k=1}^{A.cols} A_{ik} B_{kj} \]

\[ \text{Find the } C_{[i][j]} \text{ with the minimum number of multiplications} \]

**Matrix Chain**:

Given `A_1, A_2, A_3, \ldots, A_n`

Where `A_i = P_{i-1} \times P_i`

`A_1 \in \mathbb{R}`

`A_2 = P_1 \times P_2`

`A_3 = P_2 \times P_3`

Given array `P[0, \ldots, n]` of dimensions

Goal: Parenthesize product `A_1 \cdot A_2 \cdot A_3 \ldots \cdot A_n` to minimize the number of multiplications

**Example**: Let `A` be `3 \times 5` want to compute `A.B.C` two choices:

- `(AB) \cdot C` or `A(BC)`

Takes
3.5 \cdot 2 = 70
\text{\textcolor{red}{\textcircled{4}}} \text{ A(B)} \text{ will be} \\
3 \times 2
\text{\textcircled{6}} \text{ (AB)} \cdot C
\text{will take}
3 \cdot 2 \cdot 7 = 42
\text{TOTAL: } 30 + 42 = 72

TAKES 5 \cdot 2 \cdot 7 = 70
\text{\textcolor{red}{\textcircled{4}}} \text{ will be } 5 \times 7
\text{So } A(\text{BC})
\text{will take}
3 \cdot 5 \cdot 7 = 105
\text{TOTAL: } 175

WANT TO DO LESS WORK!

So, given a chain of matrices, want to figure what order to multiply in to minimize numerical operations.

Brute force? Check all parentheticalisations.

n = 3, we had 2
n = 4? (AB)(CD)  A(BC)D  AB(CD)
  /   \  /   \  /   \\
(AB) (CD) (AB) (CD) (CD) (AB)

n = 6
\textcircled{P}(u) \cdot \textcircled{P}(2)

Choose first split. Recurse on two halves.
Property: If a parenthesization is optimal (uses fewest multiplications), then the sub-chains must be optimal for their (smaller) lengths. I.e. in an optimal paren., the sub-chains are optimal.

So to count total possible arrangements:

\[ P(n) = \begin{cases} 
1 & n = 1 \\
\sum_{i=1}^{n-1} P(i) \cdot P(n-i) & n \geq 2
\end{cases} \]

= Big number \( \bigcirc \) (Catalan numbers)

\[ m[i][j] = \sum \left( \binom{\frac{n}{2}}{\frac{k}{2}} \right) \]

\( n = \text{length of chain} \)

Want to characterize the solution:

Denote \( m_{i,j} \) denote optimal \# of multiplications of the sub-chain \( A_i \cdot A_{i+1} \cdots A_j \). We want to write \( m_{i,j} \) recursively:
\[ m_{i,j} = \begin{cases} 
0 & i = j \\
\min_{i \leq k < j} \left( m_{i,k} + m_{k+1,j} + P_{i,k} \cdot P_j \right) & i < j
\end{cases} \]

Recall the ROO formulation:

\[ \Gamma_n = \max_{1 \leq i \leq n} (P_i + \Gamma_{n-i}) \]

Ex. split:

\[ (A_i \cdot A_{i+1} \cdots \cdot A_k)(A_{k+1} \cdots A_j) \]

Dim: \( P_{i-1} \times P_k \times P_j \)

To "Join" the split takes \( P_{i-1} \cdot P_k \cdot P_j \) multiplications.

Solution given by number \( M_{i,n} \).

We will track the splits along the way.

Use matrix \( M' \):

\[ M'_{ij} = m_{i,j} \]

\[ = \min \left( m_{i,2} + m_{3,j} + P_0 \cdot P_2 \cdot P_j, m_{1,3} + m_{2,j} + P_0 \cdot P_2 \cdot P_j \right) \]

\[ i = j = 3 \Rightarrow A_1 \cdot A_2 \cdot A_3 \]

\[ = P_0 \cdot P_1 \cdot P_2 \cdot P_3 \]

\[ M'_{1,3} \text{ has solution} \]

\[ \text{Solution given by number } M'_{i,n} \]

We will track the splits along the way.

Use matrix \( M' \):

\[ M'_{ij} = m_{i,j} \]

\[ = \min \left( m_{i,2} + m_{3,j} + P_0 \cdot P_2 \cdot P_j, m_{1,3} + m_{2,j} + P_0 \cdot P_2 \cdot P_j \right) \]
WE SHOULD ONLY NEED TO DO
(at most) \( O(n^2) \) work.

For other elements, less.

So total will (hopefully) be \( O(n^3) \).

Note each "diagonal" is every subchain of a particular length.

MATRIX-CHAIN \((p)\) \( // p \) is \( p[0], \ldots, p[n] \)

\( n = p \cdot \text{size} - 1 \)

CREATE \( n \times n \) MATRIX \( M \) of all 0's

CREATE \( n \times n \) MATRIX \( S \) of all 0's

\( // \) Loop all lengths from small to large.

\( \text{FOR } l = 2 \text{ to } n \) \( // l \) is length

\( \text{FOR } i = 1 \text{ to } n-l+1 \) \( // i \) is starting

\( j = i + l - 1 \)

\( m[i][j] = \infty \) \( // \) to minimize

\( \text{FOR } k = i \text{ to } j-1 \)

\( q = m[i][k] + m[k+1][j] \)

\( + p[i] \cdot p[k] \cdot p[j] \)

IF \( q < m[i][j] \)

...
\[ M(i,j) = 8 \]
\[ S[i][j] = k \]  \( \text{// REMEMBER CUT LOCATION} \)

RETURN \( m[1..n] \) \& \( S \)

PRINT-SOLN(\( S, i, j \)) \( \Rightarrow \) PRINT(\( S, 1, 4 \))

IF \( i = j \)
PRINT "A i"
ELSE
PRINT "C"
PRINT-SOLN(\( S, i, S[i][j] \))
PRINT-SOLN(\( S, S[i][j]+1, j \))
PRINT "\)"

**Proof Sketch that \( MG \) is \( \Omega(n^3) \):**
\[
\sum_{l=2}^{n} \sum_{i=1}^{n-l+1} \sum_{x=i}^{i+l-2} C = \Theta(n^3)
\]

Example: \( i = 3 \), \( l = 4 \)
\[ A_3 A_4 A_5 A_6 \]
\[ i \]
\[ j = 3 + i - 1 \]

\[ M_{i,j} = \min_{i \leq k < j} \left( M_{i,k} + M_{k+1,j} + \mathbb{P}_{i} \mathbb{P}_{k+1} \mathbb{P}_{j} \right) \]

\[ i=1 \quad j=4 \]

\[ M_{1,1} + M_{2,4} + \mathbb{P}_1 \mathbb{P}_2 \mathbb{P}_4 \]

\[ k=2 \]

\[ M_{1,2} + M_{3,4} + \mathbb{P}_1 \mathbb{P}_3 \mathbb{P}_4 \]

\[ k=3 \]

\[ M_{1,3} + M_{4,4} + \mathbb{P}_1 \mathbb{P}_4 \mathbb{P}_4 \]
$10x^2 \leq 5 \times 6, 6 \times 7, 9 \times 4$

$A_1, A_2, A_3, A_4$

$P = \begin{bmatrix} 10, 5, 6, 9, 4 \end{bmatrix}$

$S \quad M$

\[ \text{Grids with markings} \]