Chain Matrix Multiplication

Multiply n matrices to produce the result, using the fewest calculations necessary.

Recall:

\[ A \times B \neq B \times A \] - not commutative

\[ A \times B \times C = (A \times B) \times C = A \times (B \times C) \] - associative

If A is n x m and B is m x p, \( A \times B \) takes \( O(nmp) \) time.

\[
\begin{pmatrix}
A \\
& \cdot & \cdot \\
\end{pmatrix}
\times
\begin{pmatrix}
B \\
& \cdot & \cdot \\
\end{pmatrix}
=
\begin{pmatrix}
AB \\
& \cdot & \cdot \\
\end{pmatrix}
\]
Chain Matrix Multiplication

Example

\[
\begin{array}{ccc}
5 & 10 & A \\
& 3 & B \\
& 10 & C \\
& & 8 \\
& & D \\
\end{array}
\]

Cost 1

\[
\begin{align*}
AxB & \text{ costs } 5 \cdot 10 \cdot 3 = 150 \\
AxB \times C & \text{ costs } 5 \cdot 3 \cdot 8 = 120 \\
AB \times CD & \text{ costs } 5 \cdot 8 \cdot 9 = 360 \\
& \quad 630 \quad \text{Total cost (Cost 1)}
\end{align*}
\]

Cost 2

\[
\begin{align*}
AxB & \text{ costs } 5 \cdot 10 \cdot 3 = 150 \\
C \times D & \text{ costs } 3 \cdot 8 \cdot 9 = 216 \\
AB \times CD & \text{ costs } 5 \cdot 3 \cdot 9 = 135 \\
& \quad 501 \quad \text{Total cost (Cost 2)}
\end{align*}
\]

What is the lowest cost?
How many different orders are there?
Chain Matrix Multiplication

Brute Force

$(n-1)!$ different orders of calculation.
$n$ matrices, $n-1$ multiplications, any order.

Let's do better!
What is the full problem?
What are the sub problems?
What order do they need to be processed?
How many sub problems?
What is the recurrence?
What is the shape of the sub problem graph?
Chain Matrix Multiplication

Problem: Minimize cost of multiplying matrices 1 to n.

Sub Problems: Minimize cost of multiplying a subset of matrices.

How do we choose the subset?

1 - i ?

i - n ? Which one?

i - j ? Why?

any subset ?
Chain matrix multiplication

\[ C(1,n) \equiv \text{minimal cost of multiplying matrices 1 to n.} \]

\[ C(i,j) \equiv \text{minimal cost of multiplying matrices i to j} \]

\[ \text{cost}(i,k,j) \equiv \text{cost of multiplying } (M_i \times \ldots \times M_{k-1}) \times (M_k \times \ldots \times M_j) \]

\[ i < k \leq j \quad = \text{height}(M_i) \times \text{height}(M_k) \times \text{width}(M_j) \]

\[ C(i,j) = \min_{i < k \leq j} \left\{ C(i,k-1) + \text{cost}(i,k,j) + C(k,j) \right\} \]

\[ C(i,i) = 0 \quad \text{why?} \]

\[ C(i,j) = 0 \quad i > j \quad \text{why?} \]
# Chain Matrix Multiplication

## Example DAG

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $M_i = A = 5 \times 10$
- $M_2 = B = 10 \times 3$
- $M_3 = C = 3 \times 8$
- $M_4 = D = 8 \times 9$

Let $C(i,j)$ denote the minimum cost of computing the product of matrices from $A$ to $D$. Then:

\[
C(i,j) = \begin{cases} 
0 & i > j \\
0 & (i,j) \\
\min \left\{ C(i, k-1) + \cos y(i, k, j) \right\} & \text{for } k \leq j
\end{cases}
\]

- **Fill in values.**
- **Draw edge arrows.**
Chain Matrix Multiplication

Example: DAG

\[ M_1 = A = 5 \times 10 \]
\[ M_2 = B = 10 \times 3 \]
\[ M_3 = C = 3 \times 8 \]
\[ M_4 = D = 8 \times 9 \]

\[ \text{cost}(1,2) = 150 \]
\[ \text{cost}(2,3) = 240 \]
\[ \text{cost}(1,2,3) = 200 \]
\[ \text{cost}(1,3) = 120 \]
\[ \text{cost}(3,4,4) = 216 \]
\[ \text{cost}(2,3,4) = 270 \]
\[ \text{cost}(2,4,4) = 720 \]
\[ \text{cost}(1,2,4) = 450 \]
\[ \text{cost}(1,3,4) = 135 \]
\[ \text{cost}(1,4,4) = 360 \]

\[ C(1,3) = \min \left( C(1,1) + \text{cost}(1,2,3) + C(2,3), C(1,2) + \text{cost}(1,3) + C(3,4), C(1,4) + \text{cost}(1,4,4) \right) \]

\[ \begin{bmatrix} 0 & 200 & 240 \\ 150 & 120 & 0 \end{bmatrix} \]
Chain Matrix Multiplication

def CMM(M:\ll):
    for i in 1..n:
        C(i,i) = 0
    for s in 1..n-1:
        for i = 1..n-s:
            j = i+s
            C(i,j) = min_\forall k \in \ll  C(i,k-1) + cost((i,k,j)) + C(k,j)

    return C(1,n)