# Chain Matrix Multiply

Example for 4 matrices $A, B, C, D$

want to compute $A \times B \times C \times D$ most efficiently

Say $A$ is $50 \times 20$

$B$ is $20 \times 1$

$C$ is $1 \times 10$

$D$ is $10 \times 100$

Since matrix multiplication is associate we can compute:

$((A \times B) \times C) \times D$

or $(A \times B) \times (C \times D)$

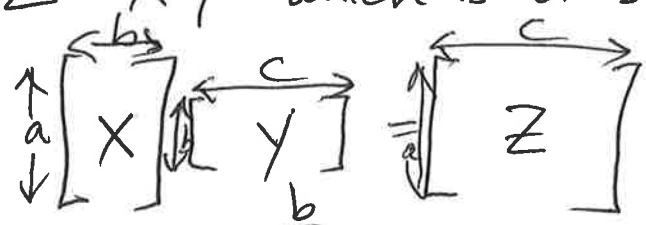
or $(A \times (B \times C)) \times D$, etc.

What's the best way?

\# of columns of $X$ = \# of rows of $Y$

Take $X$ of size $a \times b$ & $Y$ of size $b \times c$.

Let $Z = XY$ which is of size $a \times c$




$$Z_{ij} = \sum_{k=1}^{b} X_{ik} Y_{kj}$$

b multiplications & b-1 additions

Z has ac entries

So total of abc multiplications (similar # of additions)

Say cost of multiplying XY is abc.

For earlier example:

Cost for

$((A \times B) \times C) \times D$ is $\underbrace{(50)(20)(1)}_{A \times B} + \underbrace{50 \times 1 \times 10}_{(A \times B) \times C} + \underbrace{50 \times 10 \times 100}_{(A \times B \times C) \times D}$

$= 1000 + 500 + 50{,}000 = 51{,}500$

$(A \times B) \times (C \times D)$ is $(50)(20)(1) + (1)(10)(100) + (50)(1)(100)$

$= 1000 + 1000 + 5000 = 7{,}000$

$(A \times (B \times C)) \times D$ is $20 \times 1 \times 10 + 50 \times 20 \times 10 + 50 \times 10 \times 100$

$= 200 + 10{,}000 + 50{,}000 = 60{,}200$

Which ordering has the min cost?

General Problem:

For $n$ matrices $A_1, A_2, \ldots, A_n$
where $A_i$ is of size $m_{i-1} \times m_i$

What's min cost for multiplying $A_1 \times A_2 \times \cdots \times A_n$?

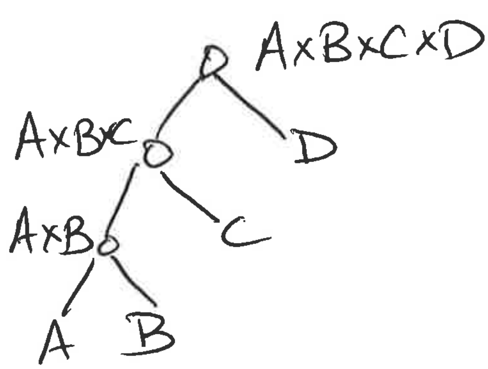
Input: $m_0, m_1, \ldots, m_n$

Goal: Min cost for computing $A_1 \times \cdots \times A_n$

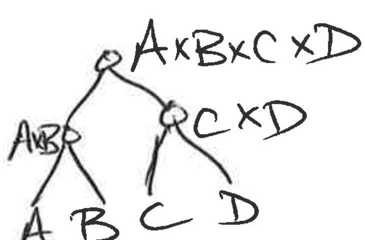
---

Graphical view:

$((A \times B) \times C) \times D$ as a binary tree




$(A \times B) \times (C \times D)$

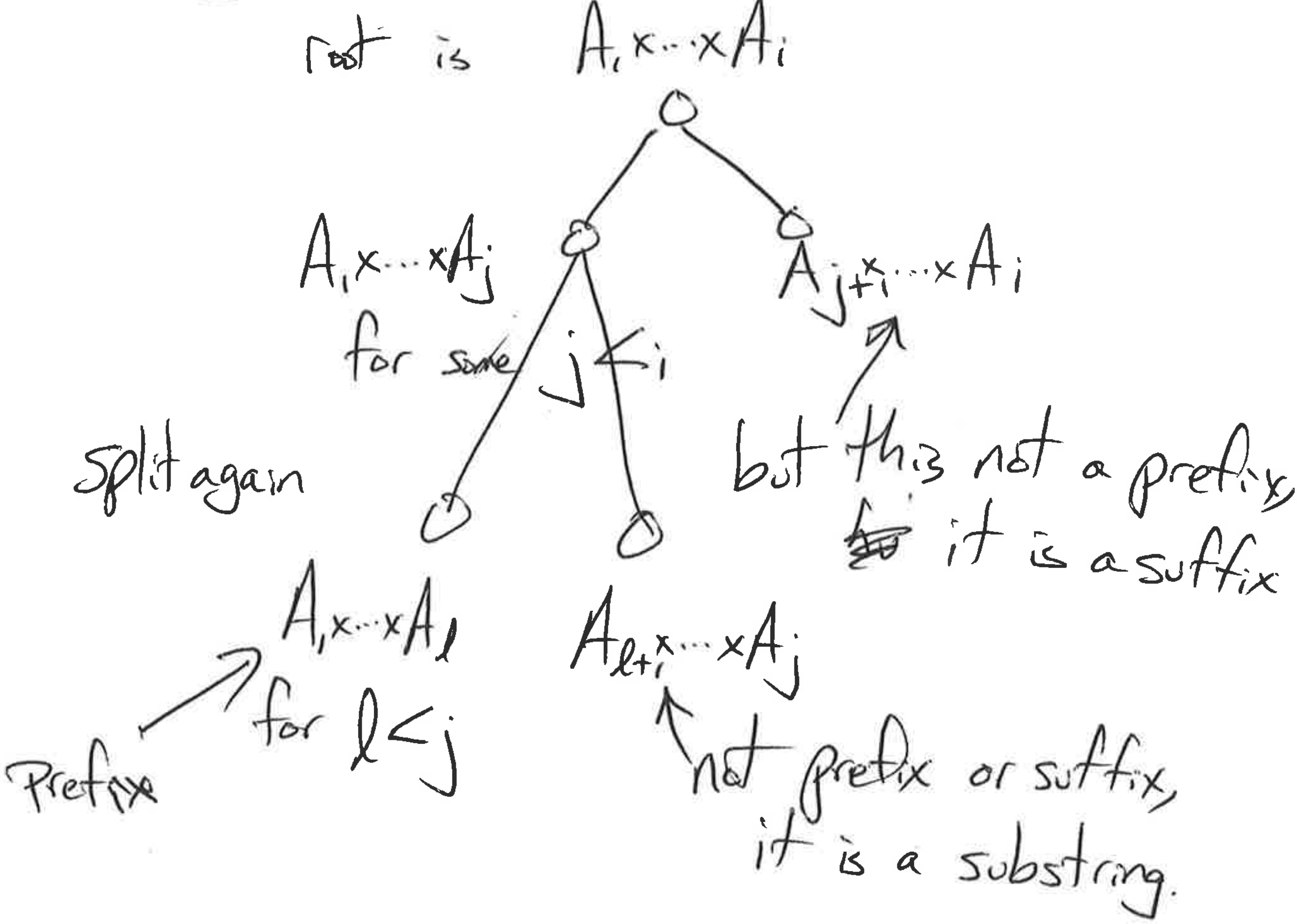



Leaves are $A_1, \ldots, A_n$. Root is $A_1 \times \cdots \times A_n$ & internal nodes are intermediate computations.

Try prefixes for subproblems:

$C(i)$ = min cost for computing $A_1 \times A_2 \times \cdots \times A_i$

looking at the tree:

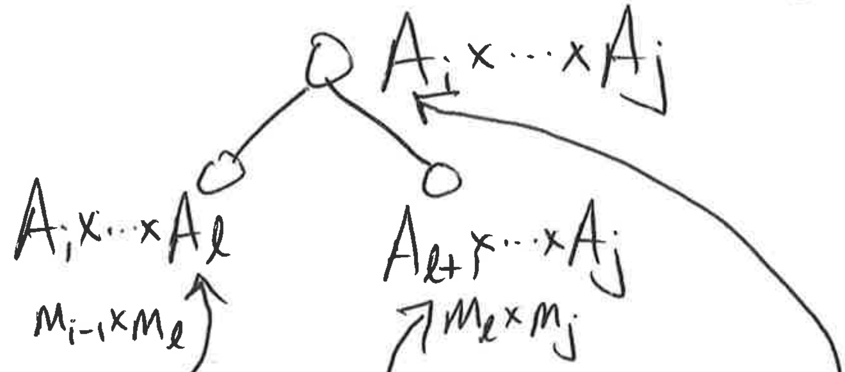


For $1 \le i \le j \le n$,

let $C(i,j) =$ min cost for computing $A_i \times \cdots \times A_j$

Base case: $C(i,i) = 0$

For $i < j$, try all $\ell$ for the split where $i \le \ell \le j-1$,



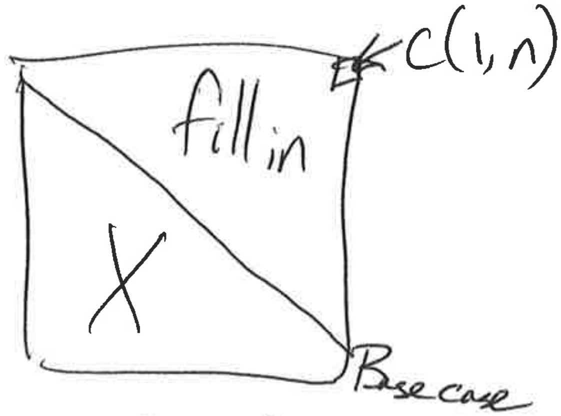

cost is $C(i,\ell) + C(\ell+1,j) + m_{i-1} m_\ell m_j$

Therefore,

$$C(i,j) = \min_{\ell} \{ C(i,\ell) + C(\ell+1,j) + m_{i-1} m_\ell m_j : i \le \ell \le j-1 \}$$

# How to fill the table?

Base case is $C(i,i)$ = diagonal




Next do $C(i, i+1)$ which uses $C(i,i)$ & $C(i+1, i+1)$

Then $C(i, i+2), \ldots$

Let $s = j - i$ = "width" of subproblem

Base case: $s = 0 \Rightarrow j = i$ is $C(i,i)$

Fill by $s = 0 \rightarrow n-1$

Goal: $C(1,n)$

# Chain Multiply $(m_0, m_1, \ldots, m_n)$:

For $i = 1 \rightarrow n$, $C(i,i) = 0$

For $s = 1 \rightarrow n-1$,

    For $i = 1 \rightarrow n-s$,

        let $j = i+s$

        $C(i,j) = \infty$

        For $l = i \rightarrow j-1$

            if $C(i,j) > \underbrace{m_{i-1} m_k m_j + C(i,l) + C(l+1,j)}$

              then $C(i,j) =$

Return $(C(1,n))$

---

Running time: 3 nested for loops of size $O(n)$ so $O(n^3)$ total time.