Revised on July 11, 2022

# MATRIX MULTIPLICATION $m(A)$ TIME COMPLEXITY ANALYSIS

by

Caleb Princewill Nwokocha

Department of Computer Science

The University of Manitoba

Winnipeg, Manitoba, Canada

May 10, 2022

CONTENT

ALGEBRA TECHNIQUE

Given two $2 \times 2$ matrices $A_1$ and $A_2$:

$$A_1 = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} e & g \\ f & h \end{pmatrix}$$

The multiplication function $m(A)$ apply the technique shown below to define dot product of $A_1$ and $A_2$.

$$m(A) = m(A_1, A_2)$$

$$m(A_1, A_2) = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot \begin{pmatrix} e & g \\ f & h \end{pmatrix}$$

Let $T = Transpose$

$$m(A_1, A_2) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^T \cdot \begin{pmatrix} e & g \\ f & h \end{pmatrix}$$

$$m(A_1, A_2) = \begin{pmatrix} \begin{pmatrix} a \\ c \end{pmatrix}^T \cdot \begin{pmatrix} e \\ f \end{pmatrix} & \begin{pmatrix} a \\ c \end{pmatrix}^T \cdot \begin{pmatrix} g \\ h \end{pmatrix} \\ \begin{pmatrix} b \\ d \end{pmatrix}^T \cdot \begin{pmatrix} e \\ f \end{pmatrix} & \begin{pmatrix} b \\ d \end{pmatrix}^T \cdot \begin{pmatrix} g \\ h \end{pmatrix} \end{pmatrix}$$

$$m(A_1, A_2) = \begin{pmatrix} (a \cdot e) + (c \cdot f) & (a \cdot g) + (c \cdot h) \\ (b \cdot e) + (d \cdot f) & (b \cdot g) + (d \cdot h) \end{pmatrix}$$

$$m(A_1, A_2) = \begin{pmatrix} ae + cf & ag + ch \\ be + df & bg + dh \end{pmatrix}$$

# $m(A)$ PSEUDOCODE

$multipy\ (matrices\ \rightarrow A_0 \cdots A_k):$

$transposed\_matrix_{m\times n}$ $O(1)$

$transposed\_matrix\_column\_elements_{m}$ $O(1)$

$highest\_column\_size = 0$ $O(1)$

$next\_matrix\_column\_elements_{m}$ $O(1)$

$temp\_product_{m\times n}$ $O(1)$

$total\_product_{m\times n}$ $O(1)$

$for\ i =\ 0\ to\ i = k - 1:$ $O(2k+2)$

Catch Index Out of Bound Exception.
$\longrightarrow break$

$for\ j = 0\ \ to\ j = row\_size(A_{i+1}) - 1:$ $O(2 \times row\_size(A_{i+1}) + 3)$

$highest\_column\_size = \frac{highest\_column\_size + column\_size(A_{i+1}, j) + |highest\_column\_size - column\_size(A_{i+1}, j)|}{2}$ $O(5)$

$temp\_product_{m\times n} = temp\_product_{row\_size(A_i) \times\ highest\_column\_size}$ $O(1)$

$transposed\_matrix_{m\times n} = transpose(A_i)$ $\sim O(a^2 + a)$

$transposed\_matrix\_column\_elements_{m} = transposed\_matrix\_column\_elements_{row\_size(transposed\_matrix_{m\times n})}$ $O(1)$

Catch Index Out of Bound Exception.
$\longrightarrow break$

$next\_matrix\_column\_elements_{m} = next\_matrix\_column\_elements_{row\_size(A_{i+1})}$ $O(2)$

$for\ l = 0\ to\ l = row\_size(temp\_product_{m\times n}) - 1:$ $O(2\ \times\ row\_size(temp\_product_{m\times n})\ +\ 2)$

$for\ p\ = 0\ to\ p\ = row\_size(transposed\_matrix\_column\_elements_{m}) - 1:$ $O(2 \times row\_size(transposed\_matrix\_column\_elements_{m}) + 2)$

$transposed\_matrix\_column\_elements_{p} = element(transposed\_matrix_{m\times n}, p, l)$ $O(1)$

$for\ q = 0\ to\ q = column\_size(temp\_product_{m\times n}, 0) - 1:$ $O(2 \times column\_size(temp\_product_{m\times n}, 0) + 2)$

$for\ r\ = 0\ to\ r\ = row\_size(next\_matrix\_column\_elements_{m}) - 1:$ $O(2\ \times\ row\_size(next\_matrix\_column\_elements_{m})\ +\ 2)$

Catch Index Out of Bound Exception.
$\longrightarrow next\_matrix\_column\_elements_{r} = 0$

$next\_matrix\_column\_elements_{r} = element(A_{i+1}, r, q)$ $O(2)$

$element(temp\_product_{m\times n}, l, q) = transposed\_matrix\_column\_elements_{m} \times next\_matrix\_column\_elements_{m}$ $\sim O(a)$

Catch Index Out of Bound Exception.
$\longrightarrow break$

$A_{i+1} = temp\_product_{m\times n}$ $O(2)$

$return: total\_product_{m\times n} = temp\_product_{m\times n}$ $O(1)$

$m(A)$ TIME COMPLEXITY

$$O(6) + [O(2k+2)][[O(2 \times row\_size(A_{i+1}) + 3)][O(5)] + O(4) + \sim O(a^2 + a)$$
$$+ [O(2 \times row\_size(temp\_product_{m \times n})$$
$$+ 2)][[O(2 \times row\_size(transposed\_matrix\_column\_elements_m) + 2)][O(1)]$$
$$+ [O(2 \times column\_size(temp\_product_{m \times n}, 0)$$
$$+ 2)][[O(2 \times row\_size(next\_matrix\_column\_elements_m) + 2)][O(2)] + \sim O(a)]]$$
$$+ \ O(2)] + O(1)$$

From the pseudocode:

$row\_size(transposed\_matrix\_column\_elements_m) = (A_i)_n$,

$row\_size(next\_matrix\_column\_elements_m) = (A_{i+1})_m$,

$column\_size(temp\_product_{m \times n}, 0) = \underbrace{arg\ max}_{n}((A_{i+1})_n)$,

$row\_size(temp\_product_{m \times n}) = (A_i)_m$,

$row\_size(A_{i+1}) = (A_{i+1})_m$,

therefore:

$$O(6) + [O(2k+2)][[O(2 \times (A_{i+1})_m + 3)][O(5)] + O(4) + \sim O(a^2 + a) + [O(2 \times (A_i)_m$$
$$+ 2)][[O(2 \times (A_i)_n + 2)][O(1)] + [O(2 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 2)][[O(2 \times (A_{i+1})_m$$
$$+ 2)][O(2)] + \sim O(a)]] + \ O(2)] + O(1)$$
$$= O(6) + [O(2k+2)][[O(2(A_{i+1})_m + 3)][O(5)] + O(4) + \sim O(a^2 + a) + [O(2(A_i)_m + 2)][[O(2(A_i)_n$$
$$+ 2)][O(1)] + [O(2 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 2)][[O(2(A_{i+1})_m + 2)][O(2)] + \sim O(a)]]$$
$$+ \ O(2)] + O(1)$$
$$= 6 + [2k+2][[2(A_{i+1})_m + 3][5] + 4 + a^2 + a + [2(A_i)_m + 2][[2(A_i)_n + 2][1]$$
$$+ [2 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 2][[2(A_{i+1})_m + 2][2] + a]] + \ 2] + 1$$
$$= 6 + [2k+2][10(A_{i+1})_m + 15 + 4 + a^2 + a + [2(A_i)_m + 2][2(A_i)_n + 2 + [2 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)$$
$$+ 2][4(A_{i+1})_m + 4 + a]] + \ 2] + 1$$

$$= 6 + [2k + 2][10(A_{i+1})_m + 19 + a^2 + a + [2(A_i)_m + 2][2(A_i)_n + 2 + 8 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m$$

$$+ 8 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 8(A_{i+1})_m + 8 + 2a] + \ 2] + 1$$

$$= 6 + [2k + 2][10(A_{i+1})_m + 19 + a^2 + a + [2(A_i)_m + 2][2(A_i)_n + 10$$

$$+ 8 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m + 8 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)$$

$$+ 8(A_{i+1})_m + 2a] + \ 2] + 1$$

$$= 6 + [2k + 2][10(A_{i+1})_m + 19 + a^2 + a + 4(A_i)_m(A_i)_n + 20(A_i)_m$$

$$+ 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m(A_i)_m + 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m$$

$$+ 2a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m + 16(A_{i+1})_m(A_i)_m + 4a(A_i)_m + 4(A_i)_n + 20$$

$$+ 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m + 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)$$

$$+ 2a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 16(A_{i+1})_m + 4a + \ 2] + 1$$

$$= 6 + [2k + 2][10(A_{i+1})_m + 41 + a^2 + 5a + 4(A_i)_m(A_i)_n + 24(A_i)_m$$

$$+ 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m(A_i)_m + 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m$$

$$+ 2a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m + 16(A_{i+1})_m(A_i)_m + 4(A_i)_n$$

$$+ 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m + 16 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)$$

$$+ 2a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 16(A_{i+1})_m] + 1$$

$$= 20(A_{i+1})_m k + 82k + 2a^2 k + 10ak + 8(A_i)_m (A_i)_n k + 48(A_i)_m k$$

$$+ 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m k$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m\, k + 32(A_{i+1})_m (A_i)_m k + 8(A_i)_n k$$

$$+ 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m k + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) k$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) k + 32(A_{i+1})_m k + 20(A_{i+1})_m + 2a^2 + 10a + 8(A_i)_m (A_i)_n$$

$$+ 48(A_i)_m + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m + 32(A_{i+1})_m (A_i)_m + 8(A_i)_n$$

$$+ 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) + 32(A_{i+1})_m + 89$$

$$\sim 20(A_{i+1})_m k + 82k + 2a^2 k + 10ak + 8(A_i)_m (A_i)_n k + 48(A_i)_m k$$

$$+ 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m k$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_i)_m\, k + 32(A_{i+1})_m (A_i)_m k + 8(A_i)_n k$$

$$+ 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m k + 32 \times \underbrace{arg\ max}_{n}((A_{i+1})_n) k$$

$$+ 4a \times \underbrace{arg\ max}_{n}((A_{i+1})_n) k + 32(A_{i+1})_m k$$

$$\sim\sim (A_{i+1})_m k + (A_i)_m (A_i)_n k + \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k$$

Assuming worst case whereby $(A_{i+1})_m = (A_i)_n = \arg\max((A_{i+1})_n) = (A_i)_m = k = n$, then:

$$(A_{i+1})_m k + (A_i)_m (A_i)_n k + \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k$$

$$= n^2 + n^3 + n^4$$

$$\sim n^4$$

$$= O(n^4)$$

This worst case shows that $(A_{i+1})_m k$ and $(A_i)_m (A_i)_n k$ have minor effect when $(A_{i+1})_m = (A_i)_n = \underbrace{arg\ max}_{n}((A_{i+1})_n) = (A_i)_m = k = n$. Also, $\underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k$ – which is $n^4$ in this case – majorly affect the running-time of $m(A)$. Hence, the worst case running-time function of $m(A)$ is $\underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m k$. The $i$ parameter of the function represents the $ith$ matrix in the set $A$, and $k$ is the number of matrices in $A$; $0 \leq i \leq k$. The running-time function of $m(A)$ can be precisely written as:

$$\sum_{i}^{k} \underbrace{arg\ max}_{n}((A_{i+1})_n)(A_{i+1})_m (A_i)_m$$

**For average case of $m(A)$:**

Let $\underbrace{arg\ max}_{n}((A_{i+1})_n)$, $(A_{i+1})_m$, $(A_i)_m$, and $k$ be a random variable $X$, such that $1 \leq X \leq n$. The probability that $X$ is any $1 \leq x \leq n$ is $\frac{1}{n}$. Therefore:

$$\text{Mean } \mu \text{ of } X,\ \mu_X = \frac{n(n+1)}{2n} = \frac{n+1}{2}$$

Consequently:

$$\mu_{\underbrace{arg\ max}_{n}((A_{i+1})_n)} = \frac{n+1}{2}$$

$$\mu_{(A_{i+1})_m} = \frac{n+1}{2}$$

$$\mu_{(A_i)_m} = \frac{n+1}{2}$$

$$\mu_k = \frac{n+1}{2}$$

$$\sum_{i}^{\mu_k} \mu_{\underbrace{arg\ max}_{n}((A_{i+1})_n)} \cdot \mu_{(A_{i+1})_m} \cdot \mu_{(A_i)_m}$$

$$= \sum_{i}^{2^{-1}(n+1)} \left(\frac{n+1}{2}\right)\left(\frac{n+1}{2}\right)\left(\frac{n+1}{2}\right)$$

$$= \left(\frac{n+1}{2}\right)\left(\frac{n+1}{2}\right)\left(\frac{n+1}{2}\right)\left(\frac{n+1}{2}\right)$$

$$\sim \frac{n^4}{16}$$

$$= n^4 \log_e 1.0644944589179$$

The average running-time of $m(A)$ is $O(n^4 \log_e 1.0645)$.

ACKNOWLEDGMENTS

I greatly thank Dr. Avery Miller for useful discussions about how to approach this analysis. Also, much thanks to my family for supporting me emotionally, financially, and otherwise, during my study and research.