UNIT 2:
Divide and Conquer

Multiplication of large Integers and Strassen's Matrix Multiplication

Multiplying two long numbers

- seek to decrease the total number of multiplications performed at the expense of a slight increase in the number of **additions.**
- exploits the divide and conquer idea.
- Applications: cryptology

Multiplication of large Integers

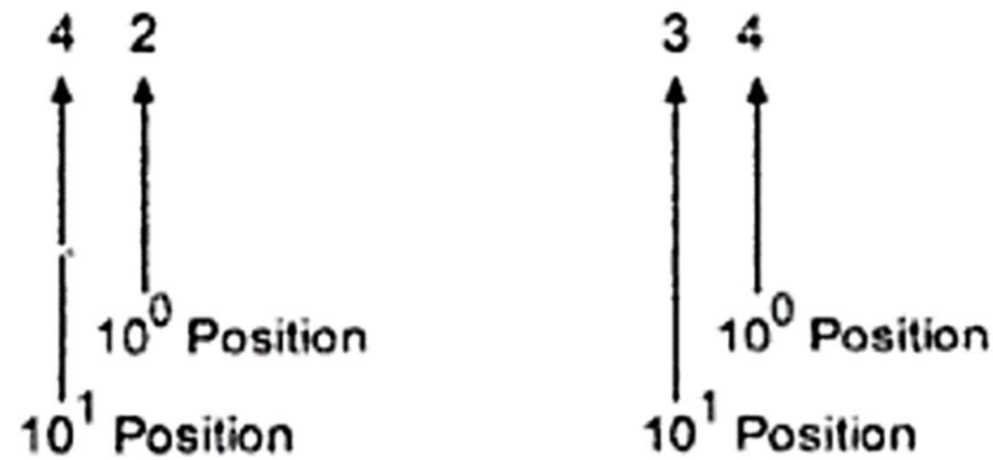
Standard algorithm for multiplying two n-digit integers:

- multiply each digit from one number with each digit from the other and then adding up the products

$$
\begin{array}{r}
25 \\
\times 63 \\
\hline
15 \\
60 \\
300 \\
+1200 \\
\hline
1575
\end{array}
$$

- Total n^2 digit multiplications.
- Divide and conquer strategy may be used to reduce the number of multiplication.

Solve : 42 X 34



i.e.

$$
\begin{aligned}
42\times 34 &= \left(4\times 10^1 + 2\times 10^0\right) * \left(3\times 10^1 + 4\times 10^0\right) \\
&= (4\times 3)10^2 + (4\times 4 + 2\times 3)10^1 + (2\times 4)10^0 \\
&= 1200 + 220 + 8 \\
&= 1428
\end{aligned}
$$

$$
\boxed{\begin{aligned} c &= a * b \\ &= c_2 10^2 + c_1 10^1 + c_0. \end{aligned}}
$$

Let us formulate this method-

$$
\begin{aligned}
c &= a * b \\
&= c_2 10^2 + c_1 10^1 + c_0,
\end{aligned}
$$

where

$c_2 = a_1 * b_1$ is the product of their first digits,
$c_0 = a_0 * b_0$ is the product of their second digits,
$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$ is the product of the sum of the a's digits and the sum of the b's digits minus the sum of c_2 and c_0.

$$
\boxed{\begin{aligned} a &= a_1 a_0 \\ b &= b_1 b_0, \end{aligned}}
$$

$$
\boxed{\begin{aligned} c_2 &= a_1 * b_1 \\ c_0 &= a_0 * b_0 \\ c_1 &= (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0) \end{aligned}}
$$

Divide and Conquer approach

$$c = a * b$$
$$= c_2 10^n + c_1 10^{n/2} + c_0$$

where

$c_2 = a_1 * b_1$ is the product of their first halves,
$c_0 = a_0 * b_0$ is the product of their second halves,
$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$ is the product of the sum of the a's halves and the sum of the b's halves minus the sum of c_2 and c_0.

$$\boxed{\begin{aligned} a &= a_1 a_0 \\ b &= b_1 b_0 \end{aligned}}$$

$$\boxed{\begin{aligned} c_2 &= a_1 * b_1 \\ c_0 &= a_0 * b_0 \\ c_1 &= (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0) \end{aligned}}$$

The recursion is stopped when n becomes one.

Let’s check our understanding

Compute 2101 * 1130 by applying the divide-and-conquer algorithm

For $2101 * 1130$:

$$\begin{aligned}
c_2 &= 21 * 11 \\
c_0 &= 01 * 30 \\
c_1 &= (21 + 01) * (11 + 30) - (c_2 + c_0) = 22 * 41 - 21 * 11 - 01 * 30.
\end{aligned}$$

For $21 * 11$:

$$\begin{aligned}
c_2 &= 2 * 1 = 2 \\
c_0 &= 1 * 1 = 1 \\
c_1 &= (2 + 1) * (1 + 1) - (2 + 1) = 3 * 2 - 3 = 3. \\
\text{So, } 21 * 11 &= 2 \cdot 10^2 + 3 \cdot 10^1 + 1 = 231.
\end{aligned}$$

For $01 * 30$:

$$\begin{aligned}
c_2 &= 0 * 3 = 0 \\
c_0 &= 1 * 0 = 0 \\
c_1 &= (0 + 1) * (3 + 0) - (0 + 0) = 1 * 3 - 0 = 3. \\
\text{So, } 01 * 30 &= 0 \cdot 10^2 + 3 \cdot 10^1 + 0 = 30.
\end{aligned}$$

For $22 * 41$:

$$\begin{aligned}
c_2 &= 2 * 4 = 8 \\
c_0 &= 2 * 1 = 2 \\
c_1 &= (2 + 2) * (4 + 1) - (8 + 2) = 4 * 5 - 10 = 10. \\
\text{So, } 22 * 41 &= 8 \cdot 10^2 + 10 \cdot 10^1 + 2 = 902.
\end{aligned}$$

Hence

$$2101 * 1130 = 231 \cdot 10^4 + (902 - 231 - 30) \cdot 10^2 + 30 = 2{,}374{,}130.$$

Divide and Conquer approach - Analysis

- Input size – N (number of digits)
- Basic operation – Multiplication
- Since multiplication of n-digit numbers requires three multiplications of n/2-digit numbers, the recurrence for the number of multiplications M (n) will be:

$$\mathbf{M(n) = 3M(n/2) \quad \text{for } n > 1,}$$

$$\mathbf{M(1) = 1.}$$

Solving it by backward substitutions for $n = 2^k$:

$M(2^k) = 3M(2^{k-1})$

$= 3[3M(2^{k-2})] = 3^2M(2^{k-2})$

$= \ldots$

$= 3^iM(2^{k-i})$

$= \ldots$

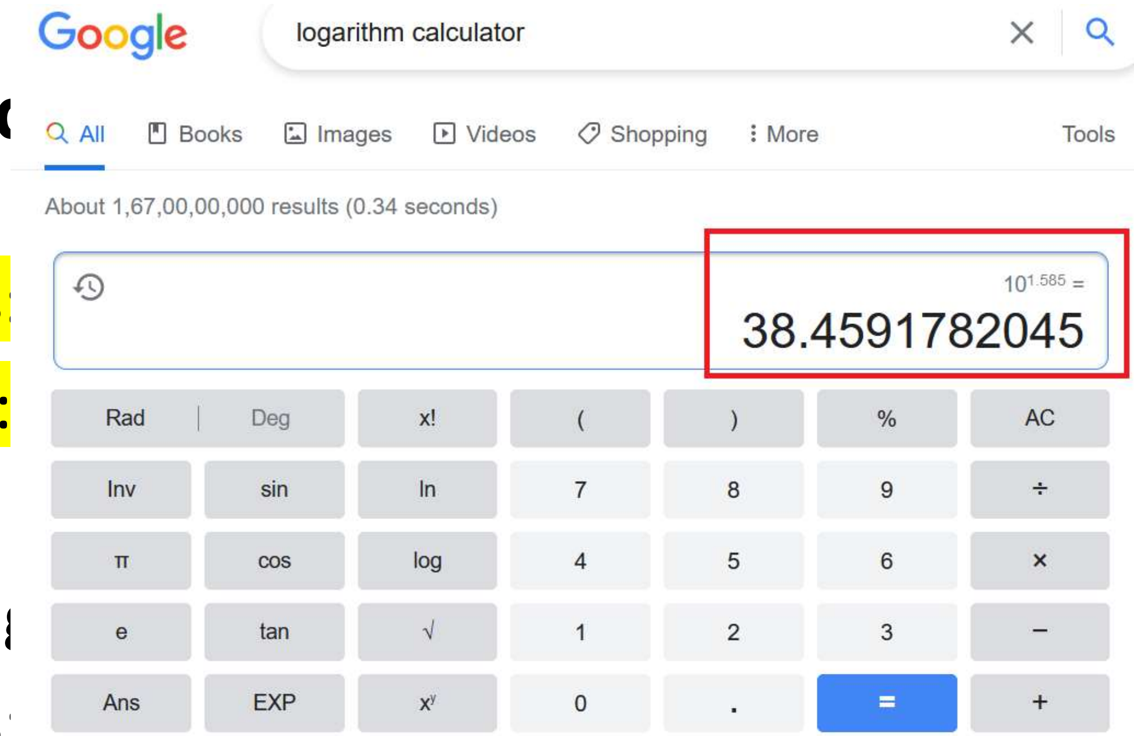
$= 3^kM(2^{k-k}) = 3^k$

Since $k = \log_2 n$

$$\boxed{M(n) = 3^{\log_2 n} = n^{\log_2 3} \approx n^{1.585}}$$

Traditional metho

- traditional methods:
- Divide and conquer:



Example 1: Multiplyin

- traditional methods:
- Divide and conquer: around 38
  - 62% decrease.

Example 2: Multiplying two 100-digit numbers

- 10,000 versus ≈ 1,445
  - 85% difference!

Strassen's Matrix Multiplication

- published by V. Strassen in 1969
- exploits the divide and conquer idea.
- can find the product C of two 2-by-2 matrices A and B with just seven multiplications as opposed to the eight required by the brute-force algorithm – $O(n^3)$
- Applications: cryptology

Strassen's Matrix Multiplication

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$
$$= \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

where

$$m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11})$$
$$m_2 = (a_{10} + a_{11}) * b_{00}$$
$$m_3 = a_{00} * (b_{01} - b_{11})$$
$$m_4 = a_{11} * (b_{10} - b_{00})$$
$$m_5 = (a_{00} + a_{01}) * b_{11}$$
$$m_6 = (a_{10} - a_{00}) * (b_{00} + b_{01})$$
$$m_7 = (a_{01} - a_{11}) * (b_{10} + b_{11}).$$

Note:
If n is not a power of two, matrices can be padded with rows and columns of zeros

Divide and Conquer approach - Analysis

Note:

To multiply two matrices of order $N > 1$, the algorithm needs to multiply seven matrices of order $N/2$ and make 18 additions of matrices of size $n/2$;

when $n = 1$, no additions are made since two numbers are simply multiplied.

- Input size – N (matrix order)
- Basic operation – Multiplication
- Number of multiplications M (n) will be:

$$\mathbf{M(n) = 7M(n/2) \quad \text{for } n > 1,}$$

$$\mathbf{M(1) = 1}$$

- Number of multiplications and additions will be:

$$\mathbf{A(n) = 7A(n/2) + 18\,(n/2)^2 \quad \text{for } n > 1,}$$

$$\mathbf{A(1) = 0}$$

Solving it by backward substitutions for $n = 2^k$:

$$M(2^k) = 7M(2^{k-1})$$
$$= 7[7M(2^{k-2})] = 7^2M(2^{k-2})$$
$$= \ldots$$
$$= 7^iM(2^{k-i})$$
$$= \ldots$$
$$= 7^kM(2^{k-k}) = 7^k$$

Since $k = \log_2 n$

Solving using Master

$$A(n) \in \Theta(n^{\log_2 7})$$

$$\boxed{M(n) = 7^{\log_2 n} = n^{\log_2 7} \approx n^{2.807}}$$

Let’s check our understanding

Apply Strassen's algorithm to compute

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$

For the matrices given, Strassen's algorithm yields the following:

$$C = \left[\begin{array}{c|c} C_{00} & C_{01} \\ \hline C_{10} & C_{11} \end{array}\right] = \left[\begin{array}{c|c} A_{00} & A_{01} \\ \hline A_{10} & A_{11} \end{array}\right] \left[\begin{array}{c|c} B_{00} & B_{01} \\ \hline B_{10} & B_{11} \end{array}\right]$$

where

$$A_{00} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}, \quad A_{01} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}, \quad A_{10} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}, \quad A_{11} = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix},$$

$$B_{00} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}, \quad B_{01} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}, \quad B_{10} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}, \quad B_{11} = \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}.$$

Therefore,

$$\begin{aligned}
M_1 &= (A_{00} + A_{11})(B_{00} + B_{11}) = \begin{bmatrix} 4 & 0 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix}, \\
M_2 &= (A_{10} + A_{11})B_{00} = \begin{bmatrix} 3 & 1 \\ 7 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix}, \\
M_3 &= A_{00}(B_{01} - B_{11}) = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -5 & 4 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix}, \\
M_4 &= A_{11}(B_{10} - B_{00}) = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}, \\
M_5 &= (A_{00} + A_{01})B_{11} = \begin{bmatrix} 3 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix}, \\
M_6 &= (A_{10} - A_{00})(B_{00} + B_{01}) = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix}, \\
M_7 &= (A_{01} - A_{11})(B_{10} + B_{11}) = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix}.
\end{aligned}$$

Accordingly,

$$\begin{aligned}
C_{00} &= M_1 + M_4 - M_5 + M_7 \\
&= \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} + \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}, \\
C_{01} &= M_3 + M_5 \\
&= \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix} + \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 1 & 9 \end{bmatrix}, \\
C_{10} &= M_2 + M_4 \\
&= \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 1 \\ 5 & 8 \end{bmatrix}, \\
C_{11} &= M_1 + M_3 - M_2 + M_6 \\
&= \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix} - \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ 7 & 7 \end{bmatrix}.
\end{aligned}$$

That is,

$$C = \begin{bmatrix} 5 & 4 & 7 & 3 \\ 4 & 5 & 1 & 9 \\ 8 & 1 & 3 & 7 \\ 5 & 8 & 7 & 7 \end{bmatrix}.$$