Divide and Conquer - Part II

Noella James

02/07/2017

1 Integer Multiplication

Input: Two n-bit numbers; a, b.
Ouput: $a \cdot b$ in bit representation

Example: $1101 \times 0111 = 1011011$. Simple algorithm with $O(n^2)$ bit operations.

1.1 Divide and Conquer - First Attempt

a $\boxed{a_1}\ \boxed{a_0}$ $\qquad a = 2^{\frac{n}{2}} a_1 + a_0$

b $\boxed{b_1}\ \boxed{b_0}$ $\qquad b = 2^{\frac{n}{2}} b_1 + b_0$

$$\begin{aligned} a \cdot b &= (2^{\frac{n}{2}} a_1 + a_0)(2^{\frac{n}{2}} b_1 + b_0) \\ &= 2^n a_1 b_1 + 2^{\frac{n}{2}}(a_1 b_0 + a_0 b_1) + a_0 b_0 \end{aligned}$$

Where $a_1 b_1, a_1 b_0, a_0 b_1, a_0 b_0$ are multiplication of $\frac{n}{2}$ bit numbers.

$a \cdot b$ $\boxed{a_1 b_1}\ \boxed{a_1 b_0 + a_0 b_1}\ \boxed{a_0 b_0}$

Overall,

$$\begin{aligned} T(n) &= 4T(\frac{n}{2}) + O(n) \\ &= O(n^2) \end{aligned}$$

Application of Masters theorem with $k = 1, b = 2, a = 4 \quad \log_b a = 2$

1.2 Karatsuba Algorithm - 1960

$$a \cdot b = 2^n a_1 b_1 + 2^{\frac{n}{2}}((a_1 + a_0)(b_1 + b_0) - a_1 b_1 - a_0 b_0) + a_0 b_0$$

$$(a_1 + a_0)(b_1 + b_0) = a_1 b_1 + a_1 b_0 + a_0 b_1 + a_0 b_0$$

Thus, we do 3 multiplications a_1b_1 , a_0b_0 , $(a_1+a_0)(b_1+b_0)$ rather than 4

$$
\begin{aligned}
T(n) &= 3T(\frac{n}{2}) + O(n) \\
&= O(n^{log3}) \\
&= O(n^{1.585})
\end{aligned}
$$

2 Matrix Multiplication

Input: Two $n \times n$ matrices, A and B.
Output: $A \cdot B$

$$
\begin{array}{cc}
\multicolumn{2}{c}{\text{A}} \\
\boxed{A_{11}} & \boxed{A_{12}} \\
\boxed{A_{21}} & \boxed{A_{22}}
\end{array}
$$

$$
\begin{array}{cc}
\multicolumn{2}{c}{\text{B}} \\
\boxed{B_{11}} & \boxed{B_{12}} \\
\boxed{B_{21}} & \boxed{B_{22}}
\end{array}
$$

$$
\begin{array}{cc}
\multicolumn{2}{c}{A \cdot B =} \\
\boxed{C_{11}} & \boxed{C_{12}} \\
\boxed{C_{21}} & \boxed{C_{22}}
\end{array}
$$

The matrices by broken by $\frac{n}{2}$
The C sub-parts can be computed as follows:

$$
\begin{aligned}
C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\
C_{12} &= A_{11}B_{12} + A_{12}B_{22} \\
C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\
C_{22} &= A_{21}B_{13} + A_{22}B_{22}
\end{aligned}
$$

2.1 First Attempt - Complexity

$$
\begin{aligned}
T(n) &= 8T(\frac{n}{2}) + O(n^2) \\
&= O(n^3)
\end{aligned}
$$

2.1.1 Applying Master's Theorem

$$
\begin{aligned}
k &= 2 \\
a &= 8 \\
b &= 2 \\
\log_2 8 &= 3 \\
k &< 3 \\
T(n) &= O(n^{\log_b a}) \\
&= O(n^3)
\end{aligned}
$$

2.2 Strassen's Algorithm (1969)

2.2.1 Computation of M's

$$
\begin{aligned}
M_1 &= (A_{11} + A_{22})(B_{12} + B_{22}) \\
M_2 &= (A_{21} + A_{22})B_{11} \\
M_3 &= A_{11}(B_{12} - B_{22}) \\
M_4 &= A_{22}(B_{21} - B_{11}) \\
M_5 &= (A_{11} + A_{12})B_{22} \\
M_6 &= (A_{21} + A_{11})(B_{11} + B_{12}) \\
M_7 &= (A_{12} - A_{22})(B_{21} + B_{22})
\end{aligned}
$$

2.2.2 Computation of C's

$$
\begin{aligned}
C_{11} &= M_1 + M_4 - M_5 + M_7 \\
C_{12} &= M_3 + M_5 \\
C_{21} &= M_2 + M_4 \\
C_{22} &= M_1 - M_2 + M_3 + M_6
\end{aligned}
$$

2.2.3 Complexity

$$
\begin{aligned}
T(n) &= 7\,T(\frac{n}{2}) + \theta(n^2) \\
&= O(n^{\log 7}) \\
&= O(n^{2.8})
\end{aligned}
$$