6.2 STRASSEN'S MATRIX-MULTIPLICATION ALGORITHM

Let A and B be two $n \times n$ matrices, where n is a power of 2. By Lemma 6.2, we can partition each of A and B into four $(n/2) \times (n/2)$ matrices and express the product of A and B in terms of these $(n/2) \times (n/2)$ matrices as:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

where

$$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21}, \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22}, \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21}, \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22}. \end{aligned} \tag{6.1}$$

If we treat A and B as 2×2 matrices, each of whose elements are $(n/2) \times (n/2$ matrices, then the product of A and B can be expressed in terms of sums and products of $(n/2) \times (n/2)$ matrices as given in (6.1). Suppose we can compute the C_{ij}'s using m multiplications and a additions (or subtractions) of the $(n/2) \times (n/2)$ matrices. By applying the algorithm recursively, we can compute the product of two $n \times n$ matrices in time $T(n)$, where for n a power of 2, $T(n)$ satisfies

$$T(n) \le mT\left(\frac{n}{2}\right) + \frac{an^2}{4}, \qquad n > 2. \tag{6.2}$$

The first term on the right of (6.2) is the cost of multiplying m pairs of $(n/2) \times (n/2)$ matrices, and the second is the cost of doing a additions, assuming $n^2/4$ time is required for each addition or subtraction of two $(n/2) \times (n/2)$ matrices. By an analysis similar to that of Theorem 2.1 with $c = 2$, we can show that as long as $m > 4$, the solution to (6.2) is bounded from above by $kn^{\log m}$ for some constant k. The form of the solution is independent of a, the number of additions. Thus if $m < 8$ we have a method asymptotically superior to the usual $O(n^3)$ method.

V. Strassen discovered a clever method of multiplying two 2×2 matrices with elements from an arbitrary ring using only seven multiplications. By using the method recursively, he was able to multiply two $n \times n$ matrices in time $O(n^{\log 7})$, which is of order approximately $n^{2.81}$.

Lemma 6.4. The product of two 2×2 matrices with elements chosen from an arbitrary ring can be computed with 7 multiplications and 1[illegible] additions/subtractions.

Proof. To compute the matrix product

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

first compute the following products.

$$
\begin{aligned}
m_1 &= (a_{12} - a_{22})(b_{21} + b_{22}),\\
m_2 &= (a_{11} + a_{22})(b_{11} + b_{22}),\\
m_3 &= (a_{11} - a_{21})(b_{11} + b_{12}),\\
m_4 &= (a_{11} + a_{12})b_{22},\\
m_5 &= a_{11}(b_{12} - b_{22}),\\
m_6 &= a_{22}(b_{21} - b_{11}),\\
m_7 &= (a_{21} + a_{22})b_{11}.
\end{aligned}
$$

Then compute the c_{ij}'s, using the formulas

$$
\begin{aligned}
c_{11} &= m_1 + m_2 - m_4 + m_6,\\
c_{12} &= m_4 + m_5,\\
c_{21} &= m_6 + m_7,\\
c_{22} &= m_2 - m_3 + m_5 - m_7.
\end{aligned}
$$

The count of operations is straightforward. The proof that the desired c_{ij}'s are obtained is a simple algebraic exercise using the laws of a ring. □

Exercise 6.5 gives a technique for computing the product of two 2×2 matrices with 7 multiplications and 15 additions.

Theorem 6.1. Two $n \times n$ matrices with elements from an arbitrary ring can be multiplied in $O(n^{\log 7})$ arithmetic operations.

Proof. First consider the case in which $n = 2^k$. Let $T(n)$ be the number of arithmetic operations needed to multiply two $n \times n$ matrices. By application of Lemma 6.4,

$$T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 \qquad \text{for} \quad n \geq 2.$$

Thus $T(n)$ is $O(7^{\log n})$ or $O(n^{\log 7})$ by an easy modification of Theorem 2.1.

If n is not a power of 2, then we embed each matrix in a matrix whose dimension is the next-higher power of 2. This at most doubles the dimension and thus increases the constant by at most a factor of 7. Thus $T(n)$ is $O(n^{\log 7})$ for all $n \geq 1$. □

Theorem 6.1 is concerned only with the functional growth rate of $T(n)$. But to know for what value of n Strassen's algorithm outperforms the usual method we must determine the constant of proportionality. However, for n not a power of 2, simply embedding each matrix in a matrix whose dimension is the next-higher power of 2 will give too large a constant. Instead, we can embed each matrix in a matrix of dimension $2^p r$ for some small value of r, and use p applications of Lemma 6.4, multiplying the $r \times r$ matrices by the conventional $O(r^3)$ method. Alternatively, we could write a more general recursive algorithm which, when n is even, would split each matrix into four submatrices as before and which, when n is odd, would first augment the dimension of the matrices by 1.