Foundation of Algorithms

Divide-and-Conquer

Contents

Strassen's Matrix Multiplication

- Problem of Matrix Multiplication
  - Time complexity of the number of multiplications in the matrix multiplication is given by $T(n) = n^3$ where n is the number of rows and columns.

Strassen's Matrix Multiplication

- Strassen's Matrix Multiplication

Suppose we want the product C of two 2 × 2 matrices, A and B. That is,

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

Strassen determined that if we let

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

the product C is given by

$$C = \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}.$$

Strassen's Matrix Multiplication

- Strassen's Matrix Multiplication (contd.)
  - To multiply two 2 × 2 matrices, Strassen's method requires seven multiplications and 18 additions/subtractions, whereas the straightforward method requires eight multiplications and 4 additions/subtractions. (worse?)
    - In the case of 2 × 2 matrices that Strassen's method is of no value.
  - However!!

$$A_{11} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1,n/2} \\ a_{21} & a_{22} \cdots a_{2,n/2} \\ & \vdots \\ a_{n/2,1} & \cdots a_{n/2,n/2} \end{bmatrix}.$$

Using Strassen's method, first we compute

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22}),$$

where our operations are now matrix addition and multiplication. In the same way, we compute M_2 through M_7. Next we compute

$$C_{11} = M_1 + M_4 - M_5 + M_7$$

Strassen's Matrix Multiplication

- Strassen's Matrix Multiplication (contd.)
  - Partitioning into submatrics in Strassen's algorithm

$$
\begin{array}{cc}
 & \leftarrow n/2 \rightarrow \\
n/2 & \left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right]
\end{array}
=
\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]
\times
\left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]
$$

Strassen's Matrix Multiplication

- Strassen's Matrix Multiplication (contd.)
  - Example
    - To multiply two matrices A and B following

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 8 & 9 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 \end{bmatrix}$$

    - Divide the matrices into submatrices

$$\overbrace{\phantom{C_{11}}}^{2}\qquad 2\left\{\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right]\right. = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{array}\right] \times \left[\begin{array}{cc|cc} 8 & 9 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ \hline 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 \end{array}\right]$$

Strassen's Matrix Multiplication

- Strassen's Matrix Multiplication (contd.)
  - Example (contd.)
    - Conquer the submatrices using Strassen's method

$$
\begin{aligned}
M_1 &= (A_{11} + A_{22}) \times (B_{11} + B_{22}) \\
&= \left(\begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ 6 & 7 \end{bmatrix} \right) \times \left(\begin{bmatrix} 8 & 9 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 9 & 1 \\ 4 & 5 \end{bmatrix} \right) \\
&= \begin{bmatrix} 3 & 5 \\ 11 & 13 \end{bmatrix} \times \begin{bmatrix} 17 & 10 \\ 7 & 9 \end{bmatrix}.
\end{aligned}
$$

    - When the matrices are sufficiently small, we multiply in the standard way

$$
\begin{aligned}
M_1 &= \begin{bmatrix} 3 & 5 \\ 11 & 13 \end{bmatrix} \times \begin{bmatrix} 17 & 10 \\ 7 & 9 \end{bmatrix} \\
&= \begin{bmatrix} 3 \times 17 + 5 \times 7 & 3 \times 10 + 5 \times 9 \\ 11 \times 17 + 13 \times 7 & 11 \times 10 + 13 \times 9 \end{bmatrix} = \begin{bmatrix} 86 & 75 \\ 278 & 227 \end{bmatrix}
\end{aligned}
$$

Strassen’s Matrix Multiplication

- Strassen’s Matrix Multiplication Algorithm

Problem: Determine the product of two $n \times n$ matrices where n is a power of 2.

Inputs: an integer n that is a power of 2, and two $n \times n$ matrices A and B.

Outputs: the product C of A and B.

```
void strassen (int n
               n × n_matrix A,
               n × n_matrix B,
               n × n_matrix& C)
{
  if (n <= threshold)
     compute C = A × B using the standard algorithm;
  else{
     partition A into four submatrices A11, A12, A21, A22;
     partition B into four submatrices B11, B12, B21, B22;
     compute C = A × B using Strassen's method;
        // example recursive call:
        // strassen(n/2, A11 + A22, B11 + B22, M1);
  }
}
```

Strassen's Matrix Multiplication

- Analysis of Strassen's Matrix Multiplication Algorithm
  - Every-case time complexity analysis of number of multiplication
    - When we have two n × n matrices with n > 1, the algorithm is called exactly seven times with an (n/2) × (n/2) matrix passed each time, and no multiplications are done at the top level.

$$T(n) = 7T\left(\frac{n}{2}\right) \qquad \text{for } n > 1,\ n \text{ a power of } 2$$
$$T(1) = 1.$$

    - Solution

$$T(n) = n^{\lg 7} \approx n^{2.81} \in \Theta\left(n^{2.81}\right)$$

Strassen’s Matrix Multiplication

- Analysis of Strassen’s Matrix Multiplication Algorithm(contd.)
  - Every-case time complexity analysis of number of additions/substractions
    - When two (n/2) × (n/2) matrices are added or subtracted, (n/2)2 additions or subtractions are done on the items in the matrices

$$T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 \qquad \text{for } n > 1, n \text{ a power of } 2$$
$$T(1) = 0.$$

    - Solution

$$T(n) = 6n^{\lg 7} - 6n^2 \approx 6n^{2.81} - 6n^2 \in \Theta\left(n^{2.81}\right)$$

Strassen's Matrix Multiplication

- When odd number of rows or columns
  - Padding with 0s
  - Comparison of two algorithms that multiply $n \times n$ matrices

	Standard Algorithm	Strassen's Algorithm
Multiplications	n^3	$n^{2.81}$
Additions/Subtractions	$n^3 - n^2$	$6n^{2.81} - 6n^2$

Arithmetic with Large Integers

- Problem
  - Suppose that we need to do arithmetic operations on integers whose size exceeds the computer's hardware capability of representing integers. If we need to maintain all the significant digits in our results, switching to a floating-point representation would be of no value.
  - Representing a large integer is to use an array of integers
    - 543.127

5	4	3	1	2	7
$S[6]$	$S[5]$	$S[4]$	$S[3]$	$S[2]$	$S[1]$

    - To represent both positive and negative integers we need only reserve the high- order array slot for the sign. We could use 0 in that slot to represent a positive integer and 1 to represent a negative integer.
  - Problem with arithmetic operation!!!

Arithmetic with Large Integers

- Solution for Arithmetic with Large Integers
  - split an n-digit integer into two integers of approximately n/2 digits.

In general, if n is the number of digits in the integer u, we will split the integer into two integers, one with $\lceil n/2 \rceil$ and the other with $\lfloor n/2 \rfloor$, as follows:

$$\underbrace{u}_{n \text{ digits}} = \underbrace{x}_{\lceil n/2 \rceil \text{ digits}} \times 10^m + \underbrace{y}_{\lfloor n/2 \rfloor \text{ digits}}$$

With this representation, the exponent m of 10 is given by

$$m = \left\lfloor \frac{n}{2} \right\rfloor .$$

If we have two n-digit integers

$$\begin{aligned} u &= x \times 10^m + y \\ v &= w \times 10^m + z, \end{aligned}$$

their product is given by

$$\begin{aligned} uv &= (x \times 10^m + y)(w \times 10^m + z) \\ &= xw \times 10^{2m} + (xz + wy) \times 10^m + yz. \end{aligned}$$

Arithmetic with Large Integers

- Solution for Arithmetic with Large Integers (contd.)
  - Example of 567,832 × 9,423,723
    - Divide

$$\underbrace{567,832}_{6\text{ digits}} = \underbrace{567}_{3\text{ digits}} \times 10^3 + \underbrace{832}_{3\text{ digits}}$$

$$\underbrace{9,423,723}_{7\text{ digits}} = \underbrace{9423}_{4\text{ digits}} \times 10^3 + \underbrace{723}_{3\text{ digits}}$$

    - Conquer

$$\begin{aligned}567,832 \times 9,423,723 &= \left(567 \times 10^3 + 832\right)\left(9423 \times 10^3 + 723\right) \\ &= 567 \times 9423 \times 10^6 + (567 \times 723 + 9423 \times 832) \\ &\quad \times 10^3 + 832 \times 723\end{aligned}$$

Arithmetic with Large Integers

- Large Integer Multiplication Algorithm

Problem: Multiply two large integers, *u* and *v*.

Inputs: large integers *u* and *v*.

Outputs: *prod*, the product of *u* and *v*.

```
large_integer prod (large_integer u, large_integer v)
{
  large_integer x, y, w, z;
  int n, m;

   n = maximum(number of digits in u, number of digits in v)
   if (u == 0 || v == 0)
      return 0;
   else if (n <= threshold)
      return u × v obtained in the usual way;
   else{
      m = ⌊n/2⌋;
      x = u divide 10^m; y = u rem 10^m;
      w = v divide 10^m; z = v rem 10^m;
      return prod(x,w) × 10^2m + (prod(x,z) + prod(w,y)) × 10^m + prod(y,z);
   }
}
```

Arithmetic with Large Integers

- Analysis of Large Integer Multiplication Algorithm
  - Then x, y, w, and z all have exactly $n/2$ digits, which means that the input size to each of the four recursive calls to prod is $n/2$. Because $m = n/2$, the linear-time operations of addition, subtraction, divide 10^m, rem 10^m, and $\times 10^m$ all have linear-time complexities in terms of n
    - Recurrence equation

$$W(n) = 4W\left(\frac{n}{2}\right) + cn \qquad \text{for } n > s,\ n \text{ a power of } 2$$

$$W(s) = 0.$$

    - solution

$$W(n) \in \Theta\left(n^{\lg 4}\right) = \Theta\left(n^2\right)$$

Arithmetic with Large Integers

- Improvement of Integer Multiplication Algorithm
  - To reduce the multiplication

$$xw, \qquad xz, \qquad yw, \qquad \text{and} \qquad yz,$$

If instead we set

$$r = (x + y)(w + z) = xw + (xz + yw) + yz,$$

then

$$xz + yw = r - xw - yz.$$

This means we can get the three values in Expression 2.4 by determining the following three values:

$$r = (x + y)(w + z), \qquad xw, \qquad \text{and} \qquad yz.$$

Arithmetic with Large Integers

- Improvement of Integer Multiplication Algorithm (contd.)
  - Algorithm

Problem: Multiply two large integers, u and v.

Inputs: large integers u and v.

Outputs: *prod2*, the product of u and v.

Arithmetic with Large Integers

- Improvement of Integer Multiplication Algorithm (contd.)

```
large_integer prod2 (large_integer u, large_integer v)
{
  large_integer x, y, w, z, r, p, q;
  int n, m;

  n = maximum(number of digits in u, number of digits in v);
  if ( u == 0 || v == 0)
     return 0;
  else if ( n <= threshold)
     return u × v obtained in the usual way;
  else{
     m = ⌊n/2⌋;
     x = u divide 10^m; y = u rem 10^m;
     w = v divide 10^m; z = v rem 10^m;
     r = prod2(x + y, w + z);
     p = prod2(x, w);
     q = prod2(y, z);
     return p × 10^(2m) + (r − p − q) × 10^m + q;
  }
}
```

Arithmetic with Large Integers

- Analysis of the Improved Large Integer Multiplication Algorithm
  - Examples of the number of digits in $x + y$ in the algorithm

n	x	y	$x+y$	Number of Digits in $x+y$
4	10	10	20	$2 = \frac{n}{2}$
4	99	99	198	$3 = \frac{n}{2} + 1$
8	1000	1000	2000	$4 = \frac{n}{2}$
8	9999	9999	19,998	$5 = \frac{n}{2} + 1$

Arithmetic with Large Integers

- Analysis of the Improved Large Integer Multiplication Algorithm (contd.)
  - Examples of the number of digits in $x + y$ in the algorithm(contd.)
    - If n is a power of 2, then x, y, w and z all have $n/2$

$$\frac{n}{2} \leq \text{ digits in } x + y \leq \frac{n}{2} + 1.$$
$$\frac{n}{2} \leq \text{ digits in } w + z \leq \frac{n}{2} + 1$$

    - Consequently

	Input Size
$prod2(x + y, w + z)$	$\frac{n}{2} \leq \text{input size} \leq \frac{n}{2} + 1$
$prod2(x, w)$	$\frac{n}{2}$
$prod2(y, z)$	$\frac{n}{2}$

Arithmetic with Large Integers

- Analysis of the Improved Large Integer Multiplication Algorithm (contd.)
  - Examples of the number of digits in $x + y$ in the algorithm(contd.)
    - Recurrence equation

$$3W\left(\frac{n}{2}\right) + cn \le W(n) \le 3W\left(\frac{n}{2} + 1\right) + cn \quad \text{for } n > s,\ n \text{ a power of } 2$$
$$W(s) = 0,$$

    - Solution

$$W(n) \in \Omega\left(n^{\log_2 3}\right)$$