Divide and Conquer
Chapter 4
Integer Multiplication
# Integer Multiplication

\[ z = x \times y \]

Each with \( n \) bits

\[
\begin{array}{cccccc}
\text{x} &=& 1 & 0 & 1 & 1 \\
\text{y} &=& 1 & 1 & 0 & 0 \\
\hline
& & 0 & 0 & 0 & 0 \\
& & 0 & 0 & 0 & 0 \\
& & 1 & 0 & 1 & 1 \\
& & 1 & 0 & 1 & 1 \\
\hline
\text{Z} &=& 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}
\]
Integer Multiplication

1. Naïve_IM(x, y)
2. $z = 0$
3. while $y > 0$
   4. if (y is odd) then $z += x$
   5. $x *= 2$
   6. $y /= 2$
7. end
8. return $z$

$T(n) = \Theta(n^2)$
D&C Integer Multiplication

\[ x \]
\[ y \]

\[ x_1 \quad x_2 \]
\[ y_1 \quad y_2 \]

\[ \frac{n}{2} \text{ bits} \quad \frac{n}{2} \text{ bits} \]
D&C Integer Multiplication

\[
x \times y = x_1 \times y_1 + x_2 \times y_2 + x_1 \times y_2 + x_2 \times y_2
\]

\[
z = \left( x_1 \frac{n}{2} + x_2 \right) \left( y_1 \frac{n}{2} + y_2 \right)
\]

\[
z = (x_1 y_1)2^n + (x_1 y_2 + x_2 y_1)2^{\frac{n}{2}} + x_2 y_2
\]

\[
T(n) = 4T \left( \frac{n}{2} \right) + n
\]

\[
T(n) = \Theta(n^2)
\]
Karatsuba’s Algorithm

\[ z = x y \]
\[ z = (x_1 y_1)2^n + (x_1 y_2 + x_2 y_1)2^{\frac{n}{2}} + x_2 y_2 \]
\[ z = (x_1 y_1)2^n + ((x_1 - x_2)(y_2 - y_1) + x_1 y_1 + x_2 y_2)2^{\frac{n}{2}} + x_2 y_2 \]

\[ T(n) = 3T\left(\frac{n}{2}\right) + n \]
\[ T(n) = \Theta(n^{\log_2 3}) \]
Example

- $x = 34 \quad x_1 = 3 \quad x_2 = 4$
- $y = 53 \quad y_1 = 5 \quad y_2 = 3$
- $n = 2$
- $z = xy$

$z = (x_1 y_1)10^2 + ((x_1 - x_2)(y_2 - y_1) + x_1 y_1 + x_2 y_2)10^2 + x_2 y_2$

- $x = (5 \cdot 3)100 + (-1 \cdot -2 + 3 \cdot 5 + 4 \cdot 3)10 + 4 \cdot 3 = 1500 + (2 + 15 + 12) \cdot 10 + 12 = 1802$
Matrix Multiplication
Section 4.2
Matrix Multiplication

\[
\begin{bmatrix}
C
\end{bmatrix}
=\begin{bmatrix}
A
\end{bmatrix}\cdot\begin{bmatrix}
B
\end{bmatrix}
\]

\[C = A \cdot B\]

\[c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj},\] assuming A, B, and C are square \(n \times n\) matrices
Simple Matrix Multiplication

\[ T(n) = \Theta(n^3) \]
D&C Matrix Multiplication

\[ A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \]

\[ C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \]

\[ \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \]

\[ C_{11} = A_{11} \cdot B_{11} + A_{12} \cdot B_{21} \]

\[ C_{12} = A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \]

\[ C_{21} = A_{21} \cdot B_{11} + A_{22} \cdot B_{21} \]

\[ C_{22} = A_{21} \cdot B_{12} + A_{22} \cdot B_{22} \]
Rec-Mat-Mult(A, B, n)

let C be a new $n \times n$ matrix

if $n == 1$

$$c_{11} = a_{11} \cdot b_{11}$$

else partition A, B, and C into $n/2 \times n/2$ submatrices

$$C_{11} = \text{Rec-Mat-Mult}(A_{11}, B_{11}, n/2) + \text{Rec-Mat-Mult}(A_{12}, B_{21}, n/2)$$
$$C_{12} = \text{Rec-Mat-Mult}(A_{11}, B_{12}, n/2) + \text{Rec-Mat-Mult}(A_{12}, B_{22}, n/2)$$
$$C_{21} = \text{Rec-Mat-Mult}(A_{21}, B_{11}, n/2) + \text{Rec-Mat-Mult}(A_{22}, B_{21}, n/2)$$
$$C_{22} = \text{Rec-Mat-Mult}(A_{21}, B_{12}, n/2) + \text{Rec-Mat-Mult}(A_{22}, B_{22}, n/2)$$

return C
Analysis of the D&C Algorithm

- \( T(n) = 8T(n/2) + \Theta(n^2) \)
- By applying the Master Theorem
  - \( a = 8, \ b = 2, \ f(n) = \Theta(n^2) \)
  - \( n^{\log_b a} = n^3 \)
  - \( f(n) = \Theta(n^2) = O(n^{3-1}) \)
- Case 1 applies
- \( T(n) = \Theta(n^{\log_b a}) = \Theta(n^3) \)
Strassen’s Algorithm

\[ A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \]
\[ C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \]

\[ S_1 = B_{12} - B_{22} \quad S_6 = B_{11} + B_{22} \]
\[ S_2 = A_{11} + A_{12} \quad S_7 = A_{12} - A_{22} \]
\[ S_3 = A_{21} + A_{22} \quad S_8 = B_{21} + B_{22} \]
\[ S_4 = B_{21} - B_{11} \quad S_9 = A_{11} - A_{21} \]
\[ S_5 = A_{11} + A_{22} \quad S_{10} = B_{11} + B_{12} \]
Strassen’s Algorithm

\[ P_1 = A_{11} \cdot S_1 \quad P_2 = S_2 \cdot B_{22} \]
\[ P_3 = S_3 \cdot B_{11} \quad P_4 = A_{22} \cdot S_4 \]
\[ P_5 = S_5 \cdot S_6 \quad P_6 = S_7 \cdot S_8 \]
\[ P_7 = S_9 \cdot S_{10} \]

\[ C_{11} = P_5 + P_4 - P_2 + P_6 \]
\[ C_{12} = P_1 + P_2 \]
\[ C_{21} = P_3 + P_4 \]
\[ C_{22} = P_5 + P_1 - P_3 - P_7 \]
Strassen’s Algo Sketch

1. Strassen(A, B, n)
2. Split A, B into quadrants
3. Compute S1, …, S10 (Matrix add/sub)
4. Compute P1, …, P7 recursively
5. Compute C1, …, C4 (Matrix add/sub)
6. Return C

\[ T(n) = 7T(n/2) + \Theta(n^2) \]
Analysis of Strassen’s Algorithm

- \( T(n) = 7T(n/2) + \Theta(n^2) \)
- By applying the Master Theorem
- \( a = 7, b = 2, f(n) = \Theta(n^2) \)
- \( n^{\log_b a} = n^{\log_2 7} = n^{2.80735...} \approx n^{2.8} \)
- \( f(n) = \Theta(n^2) = O(n^{\lg 7 - 0.8}) \)
- Case 1 applies
- \( T(n) = \Theta(n^{\log_b a}) = O(n^{2.81}) \)
Linear-time Selection
Section 9.3
Linear-time Selection

- Given an array $A$ of $n$ elements and an integer $1 \leq k \leq n$, find the $k^{th}$ smallest element in $A$
- Naïve algorithm, sort $A$ and pick the $k^{th}$ element in the sorted array $\Rightarrow \Theta(n \log n)$
- Select and remove the smallest element $k$ times $\Rightarrow \Theta(nk)$
- Another quick-sort-like algorithm
  - Pick the first element (pivot)
  - Place it in its position in the array
  - Recursively process one subarray
Quick-sort-like Algorithm

\[ A \]

\[ p_1 \]

\[ k \]
Quick-sort-like Algorithm

\[ A \]

\[ k \]

[Diagram showing partitioning of array A into two parts with indices p_1 and p_2]
Quick-sort-like Algorithm

\[ A \]

\[ k \]

\[ p_1 \]

\[ p_2 \]

\[ p_3 \]
Quick-sort-like Algorithm

$A$

$k$

$p_1$

$p_2$

$p_3$

$p_4$
Quick-sort-like Algorithm

How to choose a good pivot?

\[ T(n) = \Theta(n^2) \]
Median of Fives

\[ A = \{94, 82, 88, 12, 23, 61, 11, 13, 70, 37, 28, 31, 64, 6, 19, 32, 27, 38, 35, 21, 50, 91, 69, 57, 24, 93, 22, 43, 30, 67, 90, 48, 42, 65, 45\} \]
Median of Fives

\[ A = \{94, 82, 88, 12, 23, 61, 11, 13, 70, 37, 28, 31, 64, 6, 19, 32, 27, 38, 35, 21, 50, 91, 69, 57, 24, 93, 22, 43, 30, 67, 90, 48, 42, 65, 45\} \]

1. Partition into groups of 5
Median of Fives

A = {94, 82, 88, 12, 23, 61, 11, 13, 70, 37, 28, 31, 64, 6, 19, 32, 27, 38, 35, 21, 50, 91, 69, 57, 24, 93, 22, 43, 30, 67, 90, 48, 42, 65, 45}

2. Sort each sublist
Median of Fives

- \[ A = \{94, 82, 88, 12, 23, 61, 11, 13, 70, 37, 28, 31, 64, 6, 19, 32, 27, 38, 35, 21, 50, 91, 69, 57, 24, 93, 22, 43, 30, 67, 90, 48, 42, 65, 45\} \]

- 3. Find the median of each sublist
Median of Fives

- \( A = \{94, 82, 88, 12, 23, 61, 11, 13, 70, 37, 28, 31, 64, 6, 19, 32, 27, 38, 35, 21, 50, 91, 69, 57, 24, 93, 22, 43, 30, 67, 90, 48, 42, 65, 45\} \)
- 4. Recursively find the median of the medians
  \( M = \{82, 37, 28, 32, 57, 43, 48\} \)
- Median of medians (\( m^* \)) = 43
- Partition A around \( m^* \) and recursively process one side
Algorithm Pseudo-code

1. \textbf{SELECT}(A, n, k)
2. \textbf{if} \ (n \leq 5) \ \textbf{then} \ sort \ A \ \textbf{and return} \ k^{th} \ \textbf{element}
3. Partition \ A \ \textbf{into groups of 5}
4. \( M \leftarrow \) \textbf{Find the median of each group}
5. \( m^* = \textbf{SELECT}(M, \frac{n}{5}, \frac{n}{10}) \)
6. Partition \ A \ \textbf{around} \ m^*, \ \textbf{let it be at position} \ i
7. \textbf{if} \ (i = k) \ \textbf{then return} \ m^*
8. \textbf{if} \ (i > k) \ \textbf{then return} \ \textbf{SELECT}(A[1, i - 1], i - 1, k)
9. \textbf{if} \ (i < k) \ \textbf{then return} \ \textbf{SELECT}(A[i + 1, n], n - i, k - i)
Size of Sublist

$M$ medians of fives

$\frac{n}{5}$ group
Size of Sublist

$\frac{n}{5}$ group

$M$ medians of fives

$m^*$
Size of Sublist

\[ m \leq \frac{n}{5} \leq m^* \leq M \]

\( m^* \) medians of fives
Size of Sublist

\[ m^* \times A \]
Size of Sublist

- $\left\lfloor \frac{n}{5} \right\rfloor$ groups each of size 5
- $m^*$ is larger than half of them
- $m^*$ is larger than $\frac{n}{10}$ groups
- $m^*$ is larger than at least 3 elements in each group
- Similarly, $m^*$ is less than at least 3 elements in each group
- Size of each of the two sublists is $\left[ \frac{3}{10} n, \frac{7}{10} n \right]$
- Worst-case scenario, we prune $3n/10$ elements and recursively process $7n/10$
Recurrence Relation

\[
T(n) = \begin{cases} 
  a & ; n \leq 5 \\
  T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + an & ; n > 5 
\end{cases}
\]

- Can we apply the Master theorem?
- Let’s try recursive tree expansion
Recurrence Tree

\[ T(n) \]
Recurrence Tree

\[
\begin{align*}
T\left(\frac{n}{5}\right) & \quad an \quad T\left(\frac{7n}{10}\right)
\end{align*}
\]
Recurrence Tree

\[
\begin{align*}
\text{an} & \\
\begin{array}{cc}
\text{an} & 7\text{an} \\
5 & 10
\end{array} & \\
\begin{array}{c}
T\left(\frac{n}{25}\right) \\
T\left(\frac{7n}{50}\right)
\end{array} & \begin{array}{c}
T\left(\frac{7n}{50}\right) \\
T\left(\frac{49n}{100}\right)
\end{array}
\end{align*}
\]
Recurrence Tree

\[ a_n \]

\[
\begin{align*}
    & a_n \\
    \quad & a_n \\
    \quad & 5 \\
    \quad & T \left( \frac{n}{25} \right) \\
    \quad & T \left( \frac{7n}{50} \right) \\
\end{align*}
\]

\[
\begin{align*}
    & 7a_n \\
    \quad & 10 \\
    \quad & T \left( \frac{7n}{50} \right) \\
    \quad & T \left( \frac{49n}{100} \right) \\
    \quad & \frac{81n}{100} \\
\end{align*}
\]
Running Time

\[ T(n) = \sum_{i=0}^{d} \left( \frac{9}{10} \right)^i n = n \sum_{i=0}^{d} \left( \frac{9}{10} \right)^i \]

We do not know the depth \( d \) of the tree

\[ T(n) \leq n \sum_{i=0}^{\infty} \left( \frac{9}{10} \right)^i \leq n \left( \frac{1}{1-\frac{9}{10}} \right) \leq 10n \]

\[ T(n) = \Theta(n) \]
Proof by Induction

- We want to prove that \( T(n) = O(n) \)
- \( T(n) \leq cn \), for \( c > 0 \) and \( n \geq n_0 \)
- Base case: \( T(n) = a \) for \( n \leq 5 \)
- Setting \( c \geq a \) satisfies the base case
- Assume that \( T(n) \leq cn \) is true for all \( n \leq m \)
- We want to prove that it is true for \( n = m + 1 \)

\[
T(m + 1) = T\left(\frac{m+1}{5}\right) + T\left(\frac{7(m+1)}{10}\right) + a(m + 1)
\]

\[
T(m + 1) \leq c\left(\frac{m+1}{5}\right) + c\left(\frac{7(m+1)}{10}\right) + a(m + 1)
\]
Proof by Induction

We want to prove that
\[
c \left( \frac{m+1}{5} \right) + c \left( \frac{7(m+1)}{10} \right) + a(m+1) \leq c(m+1)
\]

\[
\frac{c}{5} + \frac{7c}{10} + a \leq c
\]

\[
\frac{c}{10} \geq a
\]

\[
c \geq 10a
\]

By setting \( c \geq 10a \), the inequality \( T(m + 1) \leq c(m + 1) \) will be true

\[
T(n) = O(n)
\]
Conclusion

- Express problems as divide and conquer algorithms
- Analyze the running time of divide and conquer algorithms
- Optimize and improve the worst-case running time of divide and conquer algorithms
- Reading: Chapter 4, Section 9.3