Integer and Matrix Multiplication

Binary Long Multiplication $\Theta(n^2)$

↳ Break into Smaller problems

e.g. 10100011 × 01100001

let $\begin{cases} a_1 = 1010,\ a_0 = 0011 \Rightarrow a = 2^{n/2} a_1 + a_0 \\ b_1 = 0110,\ b_0 = 0001 \Rightarrow b = 2^{n/2} b_1 + b_0 \end{cases}$

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

```
Multiply(A, B):
n ← size of A
if n = 1 then return A[1]·B[1]
mid ← ⌊n/2⌋
U ← Multiply(A[mid+1...n], B[mid+1...n])   // a1 b1
V ← Multiply(A[mid+1...n], B[1...mid])     // a1 b0
W ← Multiply(A[1...mid], B[mid+1...n])     // a0 b1
Z ← Multiply(A[1...mid], B[1...mid])       // a0 b0
M[1...2n] ← 0
M[1...n] ← Z  // a0 b0.
M[mid+1...] ← M[mid+1...] ⊕ V ⊕ W  // + (a1b0 + a0b1)·2^(n/2)  → << n/2 bit
M[2mid+1...] ← M[2mid+1...] ⊕ U  // a1 b1 · 2^n
return M
```

$T(n) = 4T(n/2) + \underline{\underline{n}}$ Updating Every Bit

$= 4^i T(n/2^i) + n + 2n + 4n + \cdots 2^{i-1} n$

Let $i = \log_2 n$

$T(n) = 4^{\log_2 n} + n \cdot (2^{\log_2 n} - 1)$

$= n^2 + n^2 - n = \underline{\Theta(n^2)}$

Karatsuba Multiplication

$a_1 b_0 + a_0 b_1 = \underbrace{(a_1 + a_0)(b_1 + b_0)}_{1\ \text{Extra Calculation}} - \underbrace{a_1 b_1 - a_0 b_0}_{\text{Already Calculated}}$

```
Multiply(A,B):
n ← size of A
if n = 1 then return A[1]·B[1]
mid ← ⌊n/2⌋
U ← Multiply(A[mid + 1..n], B[mid + 1..n])      //a1b1
Z ← Multiply(A[1..mid], B[1..mid])              // a0b0
A' ← A[mid + 1..n]⊕A[1..mid]                     // a1+a0
B' ← B[mid + 1..n]⊕B[1..mid]                     //b1 +b0
Y ← Multiply(A', B')                             // (a1 + a0)(b1 + b0)
M[1..2n] ← 0
M[1..n] ← M[1..n]⊕Z                              // a0b0
M[mid + 1..] ← M[mid + 1..]⊕Y ⊖ U ⊖ Z            //+(a1b0 + a0b1) ≪ n/2
M[2mid + 1..] ← M[2mid + 1..]⊕U                  // + a1b1 ≪ n
return M
```

$\Rightarrow T(n) = 3T(n/2) + n$

$= 3^{\log_2 n} + \dfrac{1 - (\frac{3}{2})^{\log_2 n}}{1 - \frac{3}{2}} \cdot n$

$= n^{\log_2 3} + 2(1 - n^{\log_2 \frac{3}{2}}) \cdot n$

$= \underline{\Theta(n^{\log_2 3})}$

Optimized Version (2019): $\Theta(n \log n)$

Matrix Multiplication

$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

Brute-force : $\Theta(n^3)$

Divide-and-Conquer

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

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

8 subproblems

$T(n) = 8T(n/2) + O(n^2)$

$\Rightarrow \underline{T(n) = O(n^3)}$ "[n elements] + [n elements]"

Strassen's Algorithm

$$\begin{cases} P_1 = A_{11} \times (B_{12} - B_{22}) \\ P_2 = (A_{11} + A_{12}) \times B_{22} \\ P_3 = (A_{21} + A_{22}) \times B_{11} \\ P_4 = A_{22} \times (B_{21} - B_{11}) \\ P_5 = (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\ P_6 = (A_{12} - A_{22}) \times (B_{21} + B_{22}) \\ P_7 = (A_{11} - A_{21}) \times (B_{11} + B_{12}) \end{cases} \Rightarrow \begin{cases} C_{11} = P_5 + P_4 - P_2 + P_6 \\ C_{12} = P_1 + P_2 \\ C_{21} = P_3 + P_4 \\ C_{22} = P_5 + P_1 - P_3 - P_7 \end{cases}$$

$T(n) = 7T(n/2) + n^2$

$= \underline{T(n^{\log_2 7})}$

Optimized Version:

Close to $\Theta(n^2)$

Master Theorem

$T(n) = aT(n/b) + f(n)$

$a \geq 1$, $b > 1$ ☆; $f(n)$: asymtotically positive polynomial function ($O(n^2) \to n^2$)

let $c = \log_b a$

$$\begin{cases} ① \text{ if } f(n) = O(n^{c-\epsilon}),\ \epsilon > 0 \\ \quad \Rightarrow T(n) = \Theta(n^c) = \Theta(n^{\log_b a}) \\ ② \text{ if } f(n) = \Theta(n^c) \\ \quad \Rightarrow T(n) = \Theta(n^c \log n) \\ ③ \text{ if } f(n) = \Omega(n^{c+\epsilon}) \\ \quad \Rightarrow T(n) = \Theta(f(n)) \end{cases}$$

⇒ for "Equalities"

Case 1: $T(n) = 3T(n/2) + n^2$

$= 3^{\log_2 n} + n^2 + \frac{3}{4} n^2 + (\frac{3}{4})^2 n^2$

$= n^{\log_2 3} + n^2 \left(1 \dfrac{1 - (\frac{3}{4})^{\log_2 n}}{1 - \frac{3}{4}}\right)$

$= n^{\log_2 3} + 4n^2 \left(1 - (\frac{3}{4})^{\log_2 n}\right)$

$= n^{\log_2 3} + 4n^2 - 4n^2 \cdot n^{\log_2 \frac{3}{4}}$

$= \Theta(n^2)$

By master theorem:

$\underline{c = \log_b a = \log_2 3}$

$f(n) = n^2 = \Omega(n^{\log_2 3 + \epsilon})$ for some ϵ (e.g. 0.1)

$\therefore T(n) = \Theta(f(n)) = \Theta(n^2)$

"此时，f(n)在函数增长时占主导"

Case 2: $T(n) = 5T(n/2) + n^2$

$= 5^{\log_2 n} + n^2 \left(1 + \frac{5}{4} + (\frac{5}{4})^2 + \cdots + (\frac{5}{4})^{\log_2 n}\right)$

$= 5^{\log_2 n} + n^2 \cdot \dfrac{(\frac{5}{4})^{\log_2 n} - 1}{\frac{1}{4}}$

$= n^{\log_2 5} + 4n^2 \cdot n^{\log_2 \frac{5}{4}} - 4n^2$

$= 5n^{\log_2 5} - 4n^2 = \Theta(n^{\log_2 5})$

$c = \log_b a = \log_2 5$

$f(n) = n^2 = O(n^{\log_2 5 - \epsilon})$ for some ϵ.

$\therefore T(n) = \Theta(n^{\log_2 5})$

Case 3: $T(n) = 4T(n/2) + n^2$

$= 4^{\log_2 n} + n^2 \cdot \log_2 n$

$= n^2 + n^2 \log_2 n = \Theta(n^2 \log n)$

$c = \log_b a = 2$

$f(n) = n^2 = \Theta(n^c)$

$\therefore T(n) = \Theta(n^c \log n) = \Theta(n^2 \log n)$

Variations:

① Master Theorem for inequalities.

$\underline{T(n) \leq aT(n/b) + f(n)}$

$$\begin{cases} f(n) = O(n^{c-\epsilon}) \Rightarrow T(n) = O(n^c) \\ f(n) = O(n^c) \Rightarrow T(n) = O(n^c \log n) \\ f(n) = \Omega(n^{c+\epsilon}) \Rightarrow T(n) = O(f(n)) \end{cases}$$

② Master Theorem when $f(n) = \Theta(n)$

$\underline{T(n) = aT(n/b) + \Theta(n)}$

$c = \log_b a$

$$\begin{cases} ① \text{ when } c > 1 \text{ then } T(n) = \Theta(n^c) \\ ② \text{ when } c = 1\quad T(n) = \Theta(n \log n) \\ ③ \text{ when } c < 1\quad T(n) = \Theta(n) \end{cases}$$

proof:

if $\log_b a \neq 1$

$T(n) = n^{\log_b a} + \Theta(n)\left(1 + \frac{a}{b} + (\frac{a}{b})^2 + \cdots + (\frac{a}{b})^{\log_b n}\right)$

$= n^{\log_b a} + \Theta(n) \dfrac{1 - (\frac{a}{b})^{(\log_b n + 1)}}{1 - \frac{a}{b}}$

$= n^{\log_b a} + \Theta(n) \dfrac{1 - (\frac{a}{b}) \cdot n^{\log_b a - 1}}{1 - \frac{a}{b}}$

$= n^{\log_b a} + \Theta(n) - \Theta(n^{\log_b a})$

$= \Theta(n) - \Theta(n^{\log_b a})$

$$\Rightarrow \begin{cases} \log_b a > 1 \Rightarrow T(n) = \Theta(n^{\log_b a}) \\ \log_b a < 1 \Rightarrow T(n) = \Theta(n) \end{cases}$$

if $\log_b a = 1$

$T(n) = n^{\log_b a} + \log_b n \cdot \Theta(n)$

$= \underline{\Theta(n \log n)}$

③ if $T(n) = n + \sum_i T(a_i n)$ $(a_i > 0,\ \sum a_i < 1)$

then $T(n) = \Theta(n)$

④ $T(n) \leq aT(n/b) + kn$

$$\begin{cases} \text{if } c > 1\quad T(n) = O(n^c) \\ \text{if } c = 1\quad T(n) = O(n \log n) \\ \text{if } c < 1\quad T(n) = O(n) \end{cases}$$