الگوریتم‌های تقسیم و غلبه

- مسئله ضرب اعداد بزرگ

- مسئله ماکزیمم جمع زیرآرایه‌ای

مسئله ضرب اعداد صحیح (بزرگ)

فرض کنید x و y دو عدد صحیح مثبت

در یک مبنای عددنویسی ثابت داده شده‌اند:

در مبنای r

$$x = (\overline{x_n x_{n-1} \cdots x_1})_r$$

$$y = (\overline{y_n y_{n-1} \cdots y_1})_r$$

فرض کنید هر دو عدد n رقمی هستند.

$\leftarrow \quad 0 \leqslant x_i \leqslant r-1$

مسئله ضرب اعداد صحیح

فرض کنید x و y دو عدد صحیح مثبت در یک مبنای عددنویسی نمایش داده شده‌اند:

در مبنای r

$$x = \left(\overline{x_n x_{n-1} \cdots x_1}\right)_r$$

$$y = \left(\overline{y_n y_{n-1} \cdots y_1}\right)_r$$

فرض کنید هر دو عدد n رقمی هستند.

توجه. نمایش یک عدد در واقع یک رشته string است.

به بیان ریاضی، ارزش مقداری x برابر است با

$$x = x_n \cdot r^{n-1} + \cdots + x_2 \cdot r + x_1$$

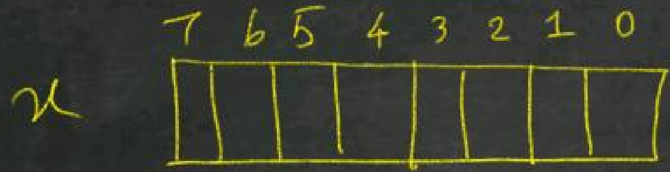
$$= \sum_{i=1}^{n} x_i \, r^{i-1}$$

هدف محاسبه $z = x * y$ است.

به قسمی که می‌خواهیم نمایش z در مبنای عددنویسی r را بدست آوریم.

توجه کنید در واقع حتی در کامپیوتر اعداد نمایش داده می‌شوند

در نمایش دیجیتال



Byte

در واقع اعداد صحیح در مبنای ۲ نمایش می‌شوند.

(کامپیوتر آنالوگ ⇐ ارزش مقداری متناظر با کمیت فیزیکی)

(کامپیوتر دیجیتال ⇐ اعداد صرفاً یک صورت نمایشی هستند)

بدیهی است مدارات محاسباتی مثل ضرب در کامپیوترها

به گونه‌ای پیاده‌سازی شده‌اند (سخت‌افزاری) که

هزینه محاسبه حاصلضرب دو عدد صحیح که قابل نمایش

در word معماری پردازنده باشند را در $O(1)$ با انجام می‌سانند

word length $= 64$ bit $\Rightarrow$ ضرب دو عدد 64 بیتی را فرض می‌گیریم

$O(1)$ است. یعنی مستقل از تعداد ارقام معنادار دو عدد است

اما فرض کنید دو عدد صحیح با تعداد ارقام n خیلی بزرگ داریم

مثلا ضرب دو عدد 4000 بیتی

سؤال ۱. حاصلضرب چگونه محاسبه (نمایش) داده می‌شود؟

8000 بیت

سؤال ۲. این الگوریتم چه هزینه‌ی محاسباتی دارد؟

مسئله ضرب اعداد صحیح

$$x = (\overline{x_n x_{n-1} \cdots x_1})_r$$

$$y = (\overline{y_n y_{n-1} \cdots y_1})_r$$

$$z = (\overline{z_{2n} z_{2n-1} \cdots z_1})_r$$

$$z = x * y$$

n در حالت کلی خیلی بزرگ

سوال در واقع تعیین ارقام عدد حاصلضرب است

$$z_i = ?$$

مثال: $x = 237$, $y = 124$

در مبنای $r = 10$

$$\begin{array}{r} 237 \\ \times\ 124 \\ \hline 948 \\ 4740 \\ +\ 23700 \\ \hline 29388 \end{array}$$

$\boxed{\text{الگوریتم ضرب مدرسه ای}} \Rightarrow T(n) = O(n^2)$

ما جدول ضرب ۱۰X۱۰ حفظ می کنیم

$$
\begin{array}{r}
1010 \\
\times\ 1001 \\
\hline
1010 \\
1010000 \\
\hline
1011010
\end{array}
$$

x

y

برای کامپیوتر کافیست جدول ضرب ۲X۲ را بداند

	0	1
0	0	0
1	0	1

عملاً معادل مدار AND

$$124 = 1\times10^2 + 2\times10^1 + 4\times10^0$$

$$237 = 2\times10^2 + 3\times10^1 + 7\times10^0$$

$$127 * 234 = \cdots + 14\times10^1 + \cdots$$

$$x_i\cdot r^{i-1} \longleftarrow x_i * y_j \longrightarrow y_j\cdot r^{j-1}$$

$$x_n \cdots \boxed{x_i} \cdots x_1$$

$$y_n \cdots y_j \cdots y_1$$

$$x_i r^{i-1} * y_j r^{j-1}$$

$$= (x_i y_j)\, r^{i+j-2}$$

یعنی $\Longrightarrow$ $k = i+j-1$ ✓✓

$$x = \sum_{k=1}^{n} x_k r^{k-1}$$

چند جمله‌ای درجهٔ n برحسب r

$$y = \sum_{k=1}^{n} y_k r^{k-1}$$

چند جمله درجهٔ n برحسب r

```
classicMultiply ( x , y )
n = x.length      //  y.length = n
Allocate a string of length 2*n for z
Initialize every digit of z by zero
for i=1 to n
  for j=1 to n
    k = i + j - 1
    z[k] = z[k] + x[i] * y[j]
for k=1 to 2*n
  if z[k] >= r then
    z[k] -= r; z[k+1]++
return z
```

الگوریتم ضرب مدرسه ای

(دقیقا معادل الگوریتم ضرب چندجمله ای)

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

سؤال: آیا الگوریتمی با هزینه کمتر وجود دارد؟

الگوریتم کاراتسوبا - الگوریتم سریع ضرب اعداد صحیح بزرگ

بر پایهٔ نگرش تقسیم و غلبه

	n	$n/2+1$	$n/2$	1
x	a		b	
y	c		d	

مثال:

$x = 235176$

$y = 421539$

$a = 235$, $b = 176$, $c = 421$, $d = 539$

$x = 1000 \times 235 + 176$, $y = 1000 \times 421 + 539$

تقسیم divide

$$\begin{cases} x = r^{n/2} \times a + b \\ y = r^{n/2} \times c + d \end{cases}$$

ترکیب combine

$$z = r^{n}(\overbrace{\underbrace{a \times c}_{1}}^{t}) + r^{n/2}(\overbrace{\underbrace{a \times d}_{2}}^{w} + \overbrace{\underbrace{b \times c}_{3}}^{v}) + \overbrace{\underbrace{b \times d}_{4}}^{u}$$

یک ایدهٔ اولیه مبتنی بر نگرش تقسیم و غلبه محاسبهٔ $\times$ به طور بازگشتی است. (Conquer غلبه)

توجه کنید طول رشته‌های a, b, c, d برابر است با $n/2$.

الگوریتم کاراتسوبا - الگوریتم سریع ضرب اعداد صحیح بزرگ

بر پایهٔ نگرش تقسیم و غلبه

	n	$n/2+1$	$n/2$	1
x	a		b	

	n	$n/2+1$	$n/2$	1
y	c		d	

تقسیم
divide

$$x = r^{n/2} \times a + b$$

$$y = r^{n/2} \times c + d$$

برای ترکیب، رشتهٔ t را به قدر n به سمت چپ شیفت می‌دهیم

رشته‌های w و v را با هم جمع، سپس حاصلجمع را به قدر $n/2$ به سمت چپ شیفت می‌دهیم

حاصلجمع دو رشتهٔ حاصل را با u جمع می‌کنیم

ترکیب
combine

$$z = r^n(\underbrace{\overbrace{a \times c}^{t}}_{1}) + r^{n/2}(\underbrace{\overbrace{a \times d}^{w}}_{2} + \underbrace{\overbrace{b \times c}^{v}}_{3}) + \underbrace{\overbrace{b \times d}^{u}}_{4}$$

یک ایدهٔ اولیه مبتنی بر نگرش تقسیم و غلبه محاسبهٔ x به طور بازگشتی است. (Conquer غلبه)

توجه کنید طول رشته‌های a, b, c, d برابر است با $n/2$.

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

```
DivideAndConquerMult(X, Y)                                    ضرب CPU
n = X.length
if n ≤ WordLengthCPU then return X*Y
[ A = Left(X, n/2);  B = Right(X, n/2)                        O(n)
[ C = Left(Y, n/2);  D = Right(Y, n/2)
[ t = DivideAndConquerMult(A, C)
[ w = DivideAndConquerMult(A, D)                              4T(n/2)
[ v = DivideAndConquerMult(B, C)
[ u = DivideAndConquerMult(B, D)
[ t = ShiftToLeftAndZeroPad(t, n)
[ wv = ShiftToLeftAndZeroPad(AddLarge(w, v), n/2)
[ return AddLarge(t, wv, u)                                   O(n)
```

```
DivideAndConquerMult(X, Y)
  n = X.length
  if n ≤ WordLengthCPU then return X * Y
  A = Left(X, n/2);   B = Right(X, n/2)
  C = Left(Y, n/2);   D = Right(Y, n/2)
  t = DivideAndConquerMult(A, C)
  w = DivideAndConquerMult(A, D)
  v = DivideAndConquerMult(B, C)
  u = DivideAndConquerMult(B, D)
  t = ShiftToLeftAndZeroPad(t, n)
  wv = ShiftToLeftAndZeroPad(AddLarge(w, v), n/2)
  return AddLarge(t, wv, u)
```

$$T(n) = 4T(n/2) + \Theta(n)$$

قضیه اساسی

$$\xRightarrow{\text{حالت اول}} \boxed{T(n) = \Theta(n^2)}$$

$$\log_2 4 = 2$$

$$n^2 > n$$

الگوریتم کاراتسوبا - الگوریتم سریع ضرب اعداد صحیح بزرگ

بر پایهٔ نگرش تقسیم و غلبه

	n … $n/2+1$	$n/2$ … 1
x	a	b
y	c	d

آیا می‌توان حاصل عبارت ⊛ را با استفاده از تنها انجام ۳ بار فراخوانی ضرب محاسبه کرد؟

$$x = r^{n/2} \times a + b$$

$$y = r^{n/2} \times c + d$$

$$z = r^{n}\left(\underbrace{\overbrace{a \times c}^{t}}_{1}\right) + r^{n/2}\left(\underbrace{\overbrace{a \times d}^{w}}_{2} + \underbrace{\overbrace{b \times c}^{v}}_{3}\right) + \underbrace{\overbrace{b \times d}^{u}}_{4} \qquad (\circledast)$$

$$x = r^{n/2} \times a + b$$

$$y = r^{n/2} \times c + d$$

$$z = r^{n}\left(\underbrace{\overbrace{a \times c}^{t}}_{1}\right) + r^{n/2}\left(\underbrace{\overbrace{a \times d}^{w}}_{2} + \underbrace{\overbrace{b \times c}^{v}}_{3}\right) + \underbrace{\overbrace{b \times d}^{u}}_{4}$$

روش کاراتسوبا -

$$f := (a+b) \underbrace{\times}_{1} (c+d) \quad \text{قرار دهید}$$

$$= a \times c + b \times c$$

$$+ a \times d + b \times d$$

$$t := a \underbrace{\times}_{2} c$$

$$u := b \underbrace{\times}_{3} d$$

آنگاه

$$z = r^{n}\, t + r^{n/2}(f - t - u) + u$$

بنابراین ۳ بار فراخوانی ضرب اعداد بزرگ به ازای $\frac{n}{2}$

۶ بار فراخوانی جمع/تفریق اعداد بزرگ به ازای $\frac{n}{2}$

۲ بار شیفت به طول n و $\frac{n}{2}$

روش کاراتسوبا -

$$z = r^{n} t + r^{n/2}(f - t - u) + u$$

$$f := \underbrace{(a+b) \times (c+d)}_{1}$$ قرار دهید

$$= a \times c + b \times c + a \times d + b \times d$$

$$t := \underbrace{a \times c}_{2}$$

$$u := \underbrace{b \times d}_{3}$$

$3T(n/2)$ { بنابراین ۳ بار فراخوانی ضرب اعداد بزرگ به ازای $n/2$

$\Theta(n)$ { ۶ بار فراخوانی جمع/تفریق اعداد بزرگ به ازای $n/2$
۲ بار شیفت به طول n و $n/2$

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

قضیه اساسی $\Longrightarrow$ حالت اول

$$T(n) = \Theta(n^{\lg 3})$$

$$\lg 3 \simeq 1.59$$