طراحی الگوریتم ها

روش تقسیم و حل(ادامه)

استاد درس: مهدی جبل عاملی

ضرب ماتریس ها

- زمان اجرا:
  - عمل اصلی ضرب: n^3
  - عمل اصلی جمع: n^3-n^2

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

ضرب ماتریس ها

- زمان اجرا:
  - عمل اصلی ضرب: n^3
  - عمل اصلی جمع: n^3-n^2

ضرب ماتریس ها

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

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

- زمان اجرا:
  - عمل اصلی ضرب: n^3
  - عمل اصلی جمع: n^3-n^2

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 - m_2 + m_3 + m_6 \end{bmatrix}$$

$$m_1 = (a_{11} + a_{22})(b_{11} + b_{22})$$

$$m_2 = (a_{21} + a_{22})\, b_{11}$$

$$m_3 = a_{11}\,(b_{12} - b_{22})$$

$$m_4 = a_{22}\,(b_{21} - b_{11})$$

$$m_5 = (a_{11} + a_{12})\, b_{22}$$

$$m_6 = (a_{21} - a_{11})(b_{11} + b_{12})$$

$$m_7 = (a_{12} - a_{22})(b_{21} + b_{22})$$

- زمان اجرا:
  - عمل اصلی ضرب: n^3
  - عمل اصلی جمع: n^3-n^2

ماتریس با ابعاد n>=2

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \quad \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

ماتریس با ابعاد n>=2

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \quad \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

$$\left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right] \times \left[\begin{array}{cc|cc} 32 & 28 & 24 & 20 \\ 31 & 27 & 23 & 19 \\ \hline 30 & 26 & 22 & 18 \\ 29 & 25 & 21 & 17 \end{array}\right]$$

ماتریس با ابعاد n>=2

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \quad \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

$$\left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right] \times \left[\begin{array}{cc|cc} 32 & 28 & 24 & 20 \\ 31 & 27 & 23 & 19 \\ \hline 30 & 26 & 22 & 18 \\ 29 & 25 & 21 & 17 \end{array}\right]$$

$$m_2 = (a_{21} + a_{22})\, b_{11}$$

$$m_2 = \left(\begin{bmatrix} 9 & 10 \\ 13 & 14 \end{bmatrix} + \begin{bmatrix} 11 & 12 \\ 15 & 16 \end{bmatrix} \right) \times \begin{bmatrix} 32 & 28 \\ 31 & 27 \end{bmatrix} = \begin{bmatrix} 20 & 22 \\ 28 & 30 \end{bmatrix} \times \begin{bmatrix} 32 & 28 \\ 31 & 27 \end{bmatrix}$$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

Matrix Strassen (n, $A[1..n, 1..n]$, $B[1..n, 1..n]$) {

if $n=1$ return $A \times B$;

Partition A into $A_{11}, A_{12}, A_{21}, A_{22}$;

Partition B into $B_{11}, B_{12}, B_{21}, B_{22}$;

Compute $C = A \times B$ (با روش استراسن) ;

return C ; }

ضرب $T(n) \longrightarrow n \times n$ تعداد ضرب‌های مورد نیاز برای ضرب دو ماتریس

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

$T(n) \rightarrow n \times n$ تعداد ضرب‌های مورد نیاز برای ضرب دو ماتریس

$$\begin{cases} T(n) = 7\,T(n/2) \\ T(1) = 1 \end{cases}$$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

$T(n) \longrightarrow$ تعداد ضرب‌های مورد نیاز برای ضرب دو ماتریس $n \times n$

$$\begin{cases} T(n) = 7\,T(n/2) \\ T(1) = 1 \end{cases} \qquad T(n) \in \theta\left(n^{\log_2 7}\right) < n^{\log_2 8} = n^3$$

- نکته جانبی:

- توجه کنید که تعداد جمع ها برای جمع دو ماتریس k*k برابر است با k^2

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

جمع $T(n) \rightarrow$ تعداد جمع های مورد نیاز برای ضرب دو ماتریس $n\times n$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A x B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A x B (با روش استراسن) ;
    return C ; }
```

$T(n) \rightarrow$ تعداد جمع های مورد نیاز برای ضرب دو ماتریس $n \times n$

$T(n) = 18\left(\frac{n}{2}\right)^2 +$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A x B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A x B (با روش استراسن) ;
    return C ; }
```

$T(n) \rightarrow$ تعداد جمع های مورد نیاز برای ضرب دو ماتریس $n \times n$

$$T(n) = \underbrace{18\left(\frac{n}{2}\right)^2} + \underbrace{7\,T\left(\frac{n}{2}\right)}$$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

$T(n) \rightarrow$ تعداد جمع های مورد نیاز برای ضرب دو ماتریس $n \times n$

$$T(n) = 18\left(n/2\right)^2 + 7T\left(n/2\right)$$

$$T(1) = 0$$

```
Matrix Strassen (n, A[1..n, 1..n], B[1..n, 1..n]) {
    if n=1 return A×B ;
    Partition A into A11, A12, A21, A22 ;
    Partition B into B11, B12, B21, B22 ;
    Compute C = A×B (با روش استراسن) ;
    return C ; }
```

$T(n) \rightarrow$ تعداد جمع های مورد نیاز برای ضرب دو ماتریس $n \times n$

$$T(n) = 18\left(\frac{n}{2}\right)^2 + 7T\left(\frac{n}{2}\right) \Rightarrow T(n) \in \Theta\left(n^{\log_2 7}\right)$$

$$T(1) = 0$$

تمرین:

- الف- با استفاده از الگوریتم استراسن، دو ماتریس ۳*۳ را در یکدیگر ضرب نمایید در صورتی که به ضرب ماتریسهای ۲*۲ نیاز داشتید به جای استراسن از ضرب معمولی ماتریس ها استفاده کنید.
- ب- با توجه به شرایط قسمت الف، تعداد ضرب ها و جمع های مورد نیاز را به صورت دقیق بیان کنید و انها را با تعداد جمع و ضرب مورد نیاز برای ضرب ماتریس های ۳*۳ به روش معمولی مقایسه و نظرتان را اعلام کنید که کدام روش بهتر است: روش الف یا روش معمولی؟

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی
  - تفریق دو عدد صحیح بزرگ n رقمی

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی
  - تفریق دو عدد صحیح بزرگ n رقمی
  - ضرب دو عدد صحیح بزرگ n رقمی

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی
  - تفریق دو عدد صحیح بزرگ n رقمی
  - ضرب دو عدد صحیح بزرگ n رقمی

$a \times 10^m$

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی
  - تفریق دو عدد صحیح بزرگ n رقمی
  - ضرب دو عدد صحیح بزرگ n رقمی

$a \times 10^m$

$a \bmod 10^m$

اعداد صحیح بزرگ

- نمایش اعداد صحیح بزرگ:
  - یک آرایه n تایی برای عدد صحیح n رقمی
- زمان اجرای اعمال روی اعداد صحیح بزرگ
  - جمع دو عدد صحیح بزرگ n رقمی
  - تفریق دو عدد صحیح بزرگ n رقمی
  - ضرب دو عدد صحیح بزرگ n رقمی

$u \times 10^m$

$u \bmod 10^m$

$u \operatorname{div} 10^m$

ضرب اعداد صحیح بزرگ

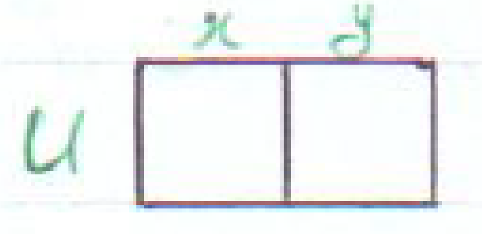
$$
\begin{array}{r}
395\,281 \\
\times\ 425\,381 \\
\hline
\end{array}
$$

ضرب اعداد صحیح بزرگ

$$\begin{array}{rr} & 395|281 \\ \times & 425|381 \\ \hline \end{array}$$

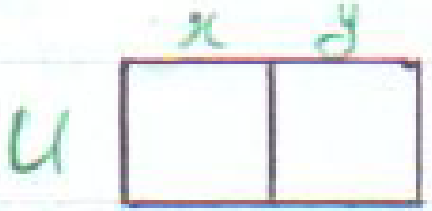
ضرب اعداد صحیح بزرگ

$$\begin{array}{r} 395|281 \\ \times\ 425|381 \\ \hline \end{array}$$



ضرب اعداد صحیح بزرگ

$$\begin{array}{r} 395|281 \\ \times\ 425|381 \\ \hline \end{array}$$

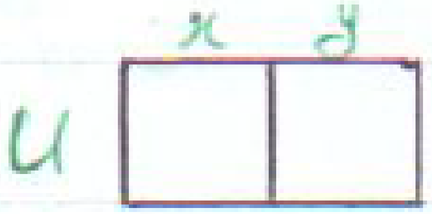


$m = \lceil n/2 \rceil$

$y = U \bmod 10^m$

ضرب اعداد صحیح بزرگ

$$\begin{array}{r} 395\,|\,281 \\ \times\ 425\,|\,381 \\ \hline \end{array}$$



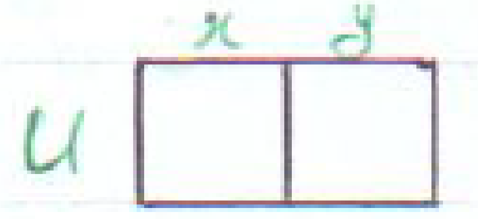
$m = \lceil n/2 \rceil$

$y = u \bmod 10^m$

$x = u \operatorname{div} 10^m$

ضرب اعداد صحیح بزرگ

$$\begin{array}{r} 395|281 \\ \times\ 425|381 \end{array}$$



$m = \lceil n/2 \rceil$

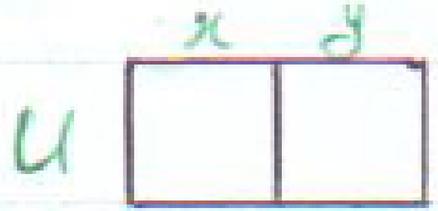
$y = u \bmod 10^m$

$x = u \operatorname{div} 10^m$

$u = x \times 10^m + y$

ضرب اعداد صحیح بزرگ

$$\begin{array}{r} 395|281 \\ \times\quad 425|381 \end{array}$$

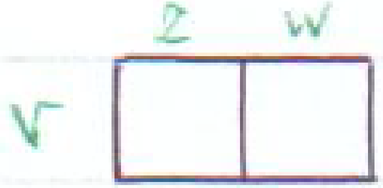


$m = \lceil n/2 \rceil$

$y = u \bmod 10^m$

$x = u \operatorname{div} 10^m$

$u = x \times 10^m + y$



$w = v \bmod 10^m$ $\quad z = v \operatorname{div} 10^m$ $\quad v = z \times 10^m + w$

ضرب اعداد صحیح بزرگ

$$uv = (x \times 10^m + y)(z \times 10^m + w)$$

$$= xz \times 10^{2m} + (xw + yz) \times 10^m + yw$$

ضرب اعداد صحیح بزرگ

```
large-int Product (large-int u, large-int v) {
    n = max { len(u), len(v) };
    if  n = 1  return  u × v ;
    m = ⌈n/2⌉ ;
    x = u div 10^m   ; y = u mod 10^m ;
    z = v div 10^m   ; w = v mod 10^m ;
    return  Product(x, z) × 10^(2m) + [Product(x, w) + Product(y, z)] × 10^m
                      + Product(y, w) ; }
```

ضرب اعداد صحیح بزرگ

```
large-int Product (large-int u, large-int v) {
    n = max { len(u), len(v) };
    if  n = 1  return  u x v ;
    m = ⌈n/2⌉ ;
    x = u div 10^m   ; y = u mod 10^m ;
    z = v div 10^m   ; w = v mod 10^m ;
    return  Product(x,z) x 10^2m + [Product(x,w) + Product(y,z)] x 10^m
                          + Product(y,w) ; }
```

$T(n) = Cn +$

ضرب اعداد صحیح بزرگ

```
large-int Product (large-int u, large-int v) {
    n = max { len(u), len(v) };
    if n=1 return u×v;
    m = ⌈n/2⌉;
    x = u div 10^m ; y = u mod 10^m ;
    z = v div 10^m ; w = v mod 10^m ;
    return Product(x,z) × 10^2m + [Product(x,w) + Product(y,z)] × 10^m
           + Product(y,w); }
```

$$T(n) = Cn + 4T(n/2)$$

ضرب اعداد صحیح بزرگ

```
large-int Product (large-int u, large-int v) {
    n = max { len(u), len(v) };
    if n = 1 return u × v;
    m = ⌈n/2⌉;
    x = u div 10^m ; y = u mod 10^m ;
    z = v div 10^m ; w = v mod 10^m ;
    return Product(x,z) × 10^(2m) + [Product(x,w) + Product(y,z)] × 10^m
            + Product(y,w); }
```

$$T(n) = Cn + 4T(n/2)$$

$$T(1) = 1$$

ضرب اعداد صحیح بزرگ

```
large-int Product (large-int u, large-int v) {
    n = max {len(u), len(v)};
    if n=1 return u×v;
    m = ⌈n/2⌉;
    x = u div 10^m ; y = u mod 10^m ;
    z = v div 10^m ; w = v mod 10^m ;
    return Product(x,z) × 10^(2m) + [Product(x,w) + Product(y,z)] × 10^m
           + Product(y,w); }
```

$T(n) = cn + 4T(n/2)$

$T(1) = 1$

$\Rightarrow T(n) \in \theta(n^2)$

بهبود زمان اجرای ضرب اعداد صحیح بزرگ

$$uv = xz \times 10^{2m} + (xw + yz) \times 10^{m} + yw$$

بهبود زمان اجرای ضرب اعداد صحیح بزرگ

$$uv = xz \times 10^{2m} + (xw + yz) \times 10^{m} + yw$$

$$(x+y)(z+w)$$

$$= xz + xw + yz + yw$$

بهبود زمان اجرای ضرب اعداد صحیح بزرگ

$$uv = xz \times 10^{2m} + (xw + yz) \times 10^{m} + yw$$

$$r = (x+y)(z+w)$$

$$= xz + (xw + yz) + yw$$

بهبود زمان اجرای ضرب اعداد صحیح بزرگ

$$uv = xz \times 10^{2m} + (xw + yz) \times 10^{m} + yw$$

$$r = (x+y)(z+w)$$

$$= xz + (xw + yz) + yw$$

$$uv = xz \times 10^{2m} + [r - (xz + yw)] \times 10^{m} + yw$$

```
large-int Product2(large-int u, large-int v) {
        if (u = 0) or (v = 0)  return 0;

    n = max { len(u), len(v) }
    if n = 1  return u x v;
    m = ⌈n/2⌉;
    x = u div 10^m ; y = u mod 10^m ;
    z = v div 10^m ; w = v mod 10^m ;
    p = Product2 (x, z);
    q = Product2 (y, w);
    r = Product2 (x+y, z+w);
    return p x 10^2m + [r - (p+q)] x 10^m + q ; }
```

$$cn + 3\,w(n/2) \leq w(n) \leq cn + 2\,w(n/2) + w(n/2 + 1)$$

$w(1) = 1$

$$\underbrace{cn + 3\,w(n/2) \leq w(n)} \leq cn + 2\,w(n/2) + w(n/2 + 1)$$

$$\Rightarrow w(n) \in O\left(n^{\log_2 3}\right)$$

$$\underbrace{cn + 3W(n/2) \leq W(n)}_{\Rightarrow W(n) \in \Theta(n^{\log_2 3})} \leq cn + 2W(n/2) + W(n/2 + 1)$$

طرف راست $\Rightarrow$ $W(n) \leq cn + 3W(n/2 + 1)$ تعریف می‌کنیم $\Leftarrow S(n) = W(n+2)$

$$\underbrace{cn + 3w(n/2) \leq w(n)} \leq cn + 2w(n/2) + w(n/2+1)$$

$$\Rightarrow w(n) \in \Omega(n^{\log_2 3})$$

طرف راست $\Rightarrow$ $w(n) \leq cn + 3w(n/2+1)$ تعریف می‌کنیم $\Leftarrow$ $S(n) = w(n+2)$

$$S(n) = w(n+2) \leq c(n+2) + 3w\left(\frac{(n+2)}{2} + 1\right)$$

$$\underbrace{cn + 3w(n/2) \leqslant w(n)}_{} \leqslant cn + 2w(n/2) + w(n/2+1)$$

$$\Rightarrow w(n) \in O\left(n^{\log_2 3}\right)$$

طرف راست $\Rightarrow$ $w(n) \leqslant cn + 3w(n/2+1)$ تعریف می‌کنیم $\Leftarrow$ $S(n) = w(n+2)$

$$S(n) = w(n+2) \leqslant c(n+2) + 3w\left((n+2)/2 + 1\right)$$

$$\leqslant c(n+2) + 3w(n/2 + 2)$$

$$\underbrace{cn + 3w(n/2) \leqslant w(n)} \leqslant cn + 2w(n/2) + w(n/2+1)$$

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

طرف راست $\Rightarrow$ $w(n) \leqslant cn + 3w(n/2+1)$ تعریف می‌کنیم $\Leftarrow S(n) = w(n+2)$

$$S(n) = w(n+2) \leqslant c(n+2) + 3w\left((n+2)/2 + 1\right)$$

$$\leqslant c(n+2) + 3w(n/2 + 2)$$

$$\leqslant c(n+2) + 3S(n/2)$$

$$S(n) \leqslant c(n+2) + 3S(n/2)$$

$$\underbrace{cn + 3\,w(n/2) \leqslant w(n)} \leqslant cn + 2\,w(n/2) + w(n/2 + 1)$$

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

طرف راست $\Rightarrow$ $w(n) \leqslant cn + 3w(n/2 + 1)$ $\quad$ تعریف می‌کنیم $\Leftarrow$ $S(n) = w(n+2)$

$$S(n) = w(n+2) \leqslant c(n+2) + 3w\left((n+2)/2 + 1\right)$$

$$\leqslant c(n+2) + 3w(n/2 + 2)$$

$$\leqslant c(n+2) + 3S(n/2)$$

$$S(n) \leqslant c(n+2) + 3S(n/2)$$

$$\Rightarrow S(n) \in O\left(n^{\log_2 3}\right)$$

$$\underbrace{cn + 3w(n/2) \leq w(n)} \leq cn + 2w(n/2) + w(n/2 + 1)$$

$$\Rightarrow w(n) \in O(n^{\log_2 3})$$

طرف راست $\Rightarrow$ $w(n) \leq cn + 3w(n/2 + 1)$ تعریف می‌کنیم $\Rightarrow S(n) = w(n+2)$

$$S(n) = w(n+2) \leq c(n+2) + 3w\left((n+2)/2 + 1\right)$$

$$\leq c(n+2) + 3w(n/2 + 2)$$

$$\leq c(n+2) + 3S(n/2)$$

$$S(n) \leq c(n+2) + 3S(n/2)$$

$$\Rightarrow S(n) \in O(n^{\log_2 3}) \qquad \Rightarrow w(n) \in O(n^{\log_2 3})$$

تمرین:

- الف- با استفاده از الگوریتم Product2، دو عدد ۳ رقمی را در یکدیگر ضرب نمایید در صورتی که به ضرب اعداد ۲ یا ۱ رقمی نیاز داشتید به جای Product2 از ضرب معمولی استفاده کنید.
- ب- با توجه به شرایط قسمت الف، تعداد ضرب های مورد نیاز را به صورت دقیق بیان کنید و انها را با تعداد ضرب مورد نیاز برای ضرب دو عدد سه رقمی به روش معمولی مقایسه و نظرتان را اعلام کنید که کدام روش بهتر است: روش الف یا روش معمولی؟

قضیه اصلی

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

- تابع $n^{\log_b a}$ را با g(n) مقایسه کنید

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

قضیه اصلی

- تابع $n^{\log_b a}$ را با g(n) مقایسه کنید

$$1)\ \text{if}\quad g(n) \in O\left(n^{\log_b a}\right) \Rightarrow T(n) \in \Theta\left(n^{\log_b a}\right)$$

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

قضیه اصلی

- تابع $n^{\log_b a}$ را با g(n) مقایسه کنید

$$1)\ \text{if}\quad g(n) \in O(n^{\log_b a}) \Rightarrow T(n) \in \theta(n^{\log_b a})$$

$$2)\ \text{if}\quad g(n) \in \Omega(n^{\log_b a}) \Rightarrow T(n) \in \theta(g(n))$$

قضیه اصلی

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

- تابع $n^{\log_b a}$ را با g(n) مقایسه کنید

1) if $g(n) \in O(n^{\log_b a}) \Rightarrow T(n) \in \theta(n^{\log_b a})$

2) if $g(n) \in \Omega(n^{\log_b a}) \Rightarrow T(n) \in \theta(g(n))$

3) if $g(n) \in \theta(n^{\log_b a}) \Rightarrow T(n) \in \theta(n^{\log_b a} \times \log n)$

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

قضیه اصلی

- تابع $n^{\log_b a}$ را با g(n) مقایسه کنید

1) if $g(n) \in O(n^{\log_b a}) \Rightarrow T(n) \in \theta(n^{\log_b a})$

2) if $g(n) \in \Omega(n^{\log_b a}) \Rightarrow T(n) \in \theta(g(n))$

3) if $g(n) \in \theta(n^{\log_b a}) \Rightarrow T(n) \in \theta(n^{\log_b a} \times \log n)$

توجه داشته باشید که در مسائل، ابتدا سومین حالت را بررسی کنید و در صورت منفی بودن بقیه حالت ها را بررسی کنید

مقدار آستانه (Threshold)

- در بعضی از مواقع، شکستن مسئله به مسائل کوچکتر برای بعضی از مقادیر کوچک **n** به صرفه نیست.

- مقدار آستانه بهینه **t**، اندازه ورودی است که برای **n<t** بهتر است شکستن انجام نشود(از الگوریتم دیگری استفاده شود) ولی برای **n>t** روش تقسیم و حل بهتر از الگوریتم دیگر است.

- نکته: مقدار آستانه تاثیری در مرتبه زمان اجرای الگوریتم های گفته شده ندارد ولی می تواند زمان اجرای واقعی آنها را کاهش دهد.

پایان