نکته: عملیات ضرب ماتریس اسپارس دارای مرتبه ی زمانی زیر است:

$O(col_B \times N_A + row_A \times N_B)$

مشکل آرایه برای ماتریس اسپارس طول ثابت آرایه است. (عدم افزایش یا کاهش طول آرایه)

ضرب دو ماتریس:
$$c[\,i\,][\,j\,] = \sum_{i=0}^{k-1} a[\,i\,][\,j\,] \times b[\,i\,][\,j\,]$$

اگر ماتریس اسپارس نباشد و مرتبه ی زمانی آن را از الگوریتم اسپارس محاسبه کنیم، $T_{(n)}$ آن تغیری نمی کند.

مرتبه ی زمانی درج یا حذف یگ عنصر از ماتریس اسپارس n است.

ماتریس مثلثی:

ماتریس مربعی که تمام درایه های بالای قطر اصلی آن صفر است را ماتریس پایین مثلثی و ماتریس مربعی که تمام درایه های پایین قطر اصلی آن صفر است را ماتریس بالا مثلثی می گوییم.

$$\begin{bmatrix} \times & \cdots & \times \\ & \ddots & \vdots \\ \text{صفر} & & \times \end{bmatrix}_{n\times n}$$

بالا مثلثی

$$\begin{bmatrix} \times & & \text{صفر} \\ \vdots & \ddots & \\ \times & \cdots & \times \end{bmatrix}_{n\times n}$$

پایین مثلثی

برای ذخیره سازی ماتریس مثلثی، فقط اعداد غیر صفر آن را در یک آرایه ی یک بعدی با $\frac{n(n+1)}{2}$ خانه ذخیره می کنیم.

مثال: روش سطری:

$$\begin{array}{cccc} A & 1 & 2 & 3 \\ \begin{matrix}1\\2\\3\end{matrix} & \multicolumn{3}{c}{\begin{bmatrix} 2 & 0 & 0 \\ 3 & 6 & 0 \\ 5 & 1 & 4 \end{bmatrix}} \end{array}$$

α = 1000

خافظه مصرفی = 2

	1001	1002	1003	1004	1005	1006
B	2	3	6	5	1	4
	1	2	3	4	5	6

A[2,2] = B[3] = 6

A[3,2] = B[5] = 1

$$B[k] = A[i,j] \Rightarrow k = \frac{i(i-1)}{2} + j$$

نکته: اگر عنصر آرایه از صفر شروع شود: $k = \frac{i(i+1)}{2} + j$

$$\text{آدرس}\, B[k] = \alpha + (\frac{i(i+1)}{2} + j) \times \text{حافظه مصرفی}$$

نکته: اگر در زبان C باشد یا آرایه از صفر شروع شود به i و j یکی اضافه می شود و از کل فرمول یکی کم می شود.

در ماتریس مثلثی تعداد عناصر غیر صفر برابر $\frac{n(n+1)}{2}$ و تعداد عناصر صفر برابر $\frac{n(n-1)}{2}$ می باشد.

مثال: روش ستونی:

	1001	1002	1003	1004	1005	1006
B	2	3	6	5	1	4
	1	2	3	4	5	6

$$\begin{array}{c|c} A & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 2 & 0 & 0 \\ 3 & 6 & 0 \\ 5 & 1 & 4 \end{bmatrix} \end{array}$$

$$A[2,2] = B[4] = 6$$

$$A[3,2] = B[5] = 1$$

$$B[k] = A[i,j] \Rightarrow k = i + (j - 1)(n - \frac{j}{2})$$

نکته: اگر عنصر آرایه از صفر شروع شود: $k = i + j(n - \frac{j+1}{2})$

به طور کلی رابطه ی k با i و j برابر است با:

	سطری	ستونی
پایین مثلثی	$k = \frac{i(i-1)}{2} + j$	$k = i + (j - 1)(n - \frac{j}{2})$
بالا مثلثی	$k = j + (i - 1)(n - \frac{i}{2})$	$k = \frac{j(j-1)}{2} + i$

مثال: چنانچه ماتریس به صورت قطری ذخیره شود مرتبه ی زمانی آن برای عملیات جمع، ضرب و ترانهاده بدتر می شود یا بهتر؟

$$
\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & \cdots & n \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ \vdots \\ n \end{matrix} & \begin{bmatrix} a_{11} & & & \text{صفر} & \\ & a_{22} & & & \\ & & a_{33} & & \\ & \text{صفر} & & \ddots & \\ & & & & a_{nn} \end{bmatrix}
\end{array}
$$

0	n	n	n
1	1	1	a_{11}
2	2	2	a_{22}
3	3	3	a_{33}
⋮	⋮	⋮	⋮
n	n	n	a_{nn}

روش معمولی : حافظه مصرفی $\times\ n \times n$

روش اسپارس : حافظه مصرفی $\times\ (n + 1) \times 3$

حافظه مصرفی $\times\ n^2 >$ حافظه مصرفی $\times\ (n + 1) \times 3$

$n^2 - 3n + 3 > 0 \Rightarrow n \geq 4$

در ماتریس های تک قطری، در کران بالای n روش اسپارس بهتر است و در کران پایین n روش معمولی بهتر است.

ماتریس سه قطری:

یک ماتریس مربعی n×n می باشد که درایه های غیر صفر آن روی قطر اصلی بلافاصله بالا و پایین قطر اصلی ظاهر می شوند.

مثال: ماتریس سه قطری 4×4 زیر را به صورت سطری در یک آرایه ی یک بعدی ذخیره نمایید.

$$
\begin{array}{c|c}
A & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 6 & 37 & 0 & 0 \\ 1 & 3 & 7 & 0 \\ 0 & 2 & 4 & 83 \\ 0 & 0 & 9 & 5 \end{bmatrix}
\end{array}
$$

	1	2	3	4	5	6	7	8	9	10
B	12	20	26	30	45	65	68	70	75	92

مثال: ماتریس سه قطری 4×4 زیر را به صورت سطری در یک آرایه ی یک بعدی ذخیره نمایید.

$$
\begin{array}{c|c}
A & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 10 & 5 & 0 & 0 & 0 \\ 5 & 15 & 2 & 0 & 0 \\ 0 & 16 & 35 & 1 & 0 \\ 0 & 0 & 36 & 7 & 8 \\ 0 & 0 & 0 & 90 & 6 \end{bmatrix}
\end{array}
$$

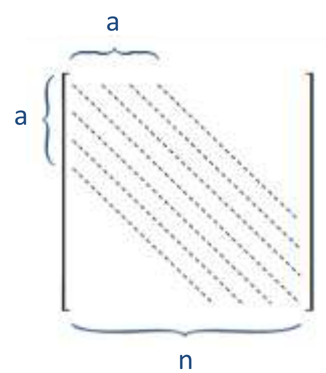
	1	2	3	4	5	6	7	8	9	10	11	12	13
B	10	5	5	15	16	2	35	36	1	7	90	8	6

ستونی	سطری	
k = i + 2j - 2	k = 2i + j - 2	سه قطری

نکته: اگر در زبان C باشد یا آرایه از صفر شروع شود به i و j یکی اضافه می شود و از کل فرمول یکی کم می شود.

نکته: تعداد عناصر غیر صفر ماتریس سه قطری برابر است با 3n-2 و تعداد عناصر صفر ماتریس سه قطری برابر است با n^2-3n+2

در ماتریس l قطری n×n تعداد $nl-(\frac{l^2-1}{4})$ غیر صفر وجود دارد.



$$l = 2a - 1 \Rightarrow a = \frac{l+1}{2}$$

تعداد عناصر غیر صفر $= n^2 - [n^2 - 2na + a^2 + n - a]$

$$= n^2 - n^2 + 2na + a^2 - n + a = n(2a - 1) - (a^2 - a)$$

$$= nl - [(\frac{l+1}{2})^2 - (\frac{l+1}{2})] = nl - (\frac{l^2 \;\; 2l + 1}{4} - \frac{2l + 2}{4})$$

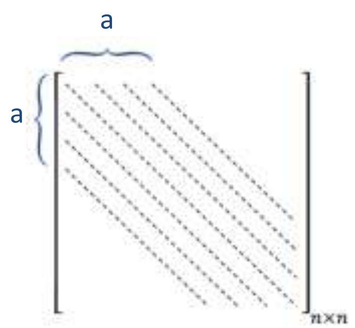
$$= nl - \frac{l^2 - 1}{4}$$

تعداد عناصر صفر $= 2\left[\frac{(n-a)(n-a+1)}{2}\right] = (n - a)^2 + (n - a) = n^2 - 2na + a^2 + n - a$

مثال:

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 3 & 3 & 7 & 0 \\ 2 & 8 & 1 & 8 & 3 \\ 3 & 1 & 7 & 4 & 3 \\ 4 & 0 & 2 & 6 & 7 \end{array}$$

$A_{4,3}$



$A_{n,a}$

تعداد قطر های پایین قطر اصلی = a-1

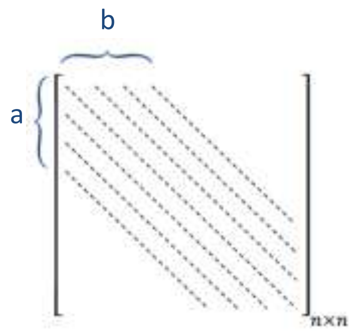
تعداد قطر های بالای قطر اصلی = a-1

تمرین۱: اگر ذخیره سازی از پایین قطر باشد رابطه ی آن را بدست آورید.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	8	7	6	3	1	4	7	3	8	3	7	3

تمرین۲: تمرین قبل را تعمیم دهید.

ضرب و جمع و ترانهاده با این ماتریس چگونه است؟ (به روش هورویس)



$A_{n,a,b}$

a-1 = تعداد قطر های پایین قطر اصلی

b-1 = تعداد قطر های بالای قطر اصلی

ماتریس متقارن:

ماتریس A را ماتریس متقارن گویند اگر `A[i,j]=A[j,i]`. در این ماتریس تنها ذخیره ی آن دسته از عناصر مورد نیاز است که بر روی قطر و پایین قطر (یا بالای قطر) قرار دارند.

برای ذخیره ی دو ماتریس، اول از ماتریس اول، پایین قطر اصلی و از ماتریس دوم، بالای قطر اصلی در نظر گرفته می شود.

$$\begin{array}{cc} A & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 5 \\ 4 & 5 & 6 \end{bmatrix} \end{array} \qquad \begin{array}{cc} B & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 7 & 8 & 9 \\ 8 & 10 & 11 \\ 9 & 11 & 12 \end{bmatrix} \end{array}$$

$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 7 & 8 & 9 \\ 2 & 3 & 10 & 11 \\ 4 & 5 & 6 & 12 \end{bmatrix} \end{array}$$

بنابراین A و B را می توان در آرایه ی n×n+1 خانه ای ذخیره کرد که در آن:

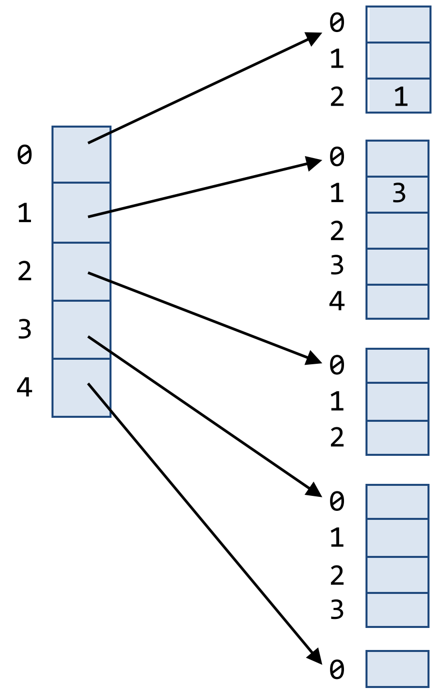
```
c[i,j] = A[i,j]          i ≥ j

c[i,j] = B[i,j-1]        i < j
```

ماتریس پله ای:

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
\begin{array}{c}1\\2\\3\\4\\5\end{array} & \multicolumn{5}{c}{\begin{bmatrix} 1 & 1 & 2 & 0 & 0 \\ 1 & 3 & 1 & 4 & 6 \\ 2 & 6 & 1 & 0 & 0 \\ 3 & 2 & 7 & 8 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}}
\end{array}
$$

```
int* A[5];
A[0] = new int[3];
A[1] = new int[5];
A[2] = new int[3];
A[3] = new int[4];
A[4] = new int[1];
A[0][2] = 1;
A[0][4] = 3;
```



چند جمله ای:

ذخیره سازی با آرایه ی یک بعدی

ذخیره سازی با آرایه ی دو بعدی

$f(x) = -6x^4 + 2x^2 + 3$

```
struct f
{
     int exp;
     Float coef;
}

f x[3];
```

	0	1	2	3	4
f(x)	3	0	2	0	-6

	0	1	2
Coef	-6	2	3
exp	4	2	0

جمع چند جمله ای ها:

$$f(x) = -6x^4 + 2x^2 + 3$$

$$g(x) = x^5 + 6x^4 + 7x^3 + x$$

$$c = \sum_{i=0}^{\max(n,m)} (a_i + b_i)$$

	0	1	2	3	4
a	3	0	2	0	-6

	0	1	2	3	4	5
b	0	1	0	7	6	1

$\Rightarrow$

	0	1	2	3	4	5
c	3	1	2	7	0	1

$$O(m+n)$$

ضرب چند جمله ای ها:

$$n \rightarrow a(x) = 2x^2 + 3$$

$$m \rightarrow b(x) = -x + 4$$

$$n + m \rightarrow c(x) = a(x) * b(x)$$

	0	1	2
a	3	0	2

	0	1
b	4	-1

$\Rightarrow$

	0	1	2	3
c	12	-3	8	-2

$$c(x) = -2x^3 + 8x^2 - 3x$$

$$c[i+j] = \sum_{i=0}^{n} \sum_{j=0}^{m} a[i] * b[j]$$

کد:

```
for (int i = 0; i <= n; i++)
    for (int j = 0; j <= mbox; j++)
        c[i + j] += a[i] * b[j];
```

$n \rightarrow a(x) = x^5 + x^3 + 2x^2 + 3$

$m \rightarrow b(x) = -2x^2 - x + 4$

	a	b		
S	0	4		
E	3	6		

S_a E_a S_b E_b

0	1	2	3	4	5	6	7	8	9	10		
3	2	3	1	4	-1	-2	7	-1	3	1		
0	2	3	5	0	1	2	0	1	3	5		

$$c(x) = x^5 + 3x^3 - x + 7$$

$$O(n + m)$$

	بهترین	بدترین
آرایه یک بعدی	Max(n , m)	Max(n , m)
آرایه دو بعدی	Max(n , m)	(n + m)