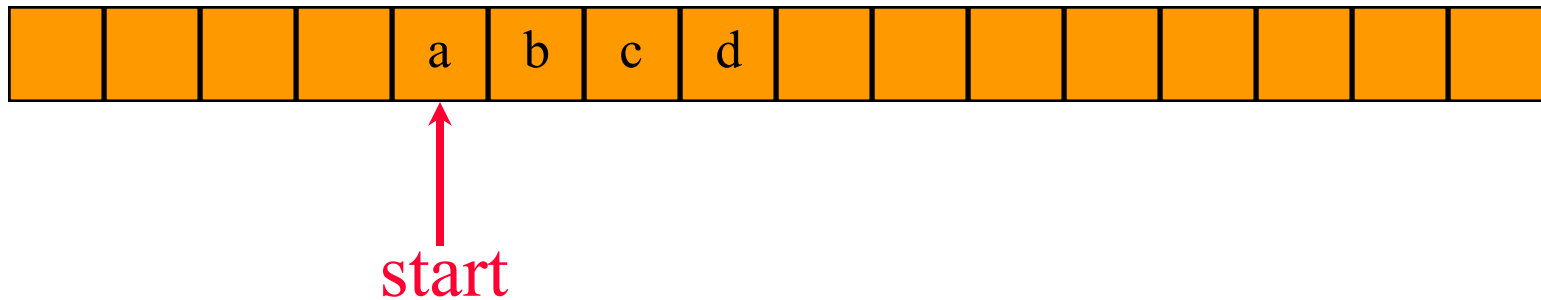


# Массив



# Java, C, C++ -д НЭГ ХЭМЖЭЭСТ МАССИВЫГ дүрслэх

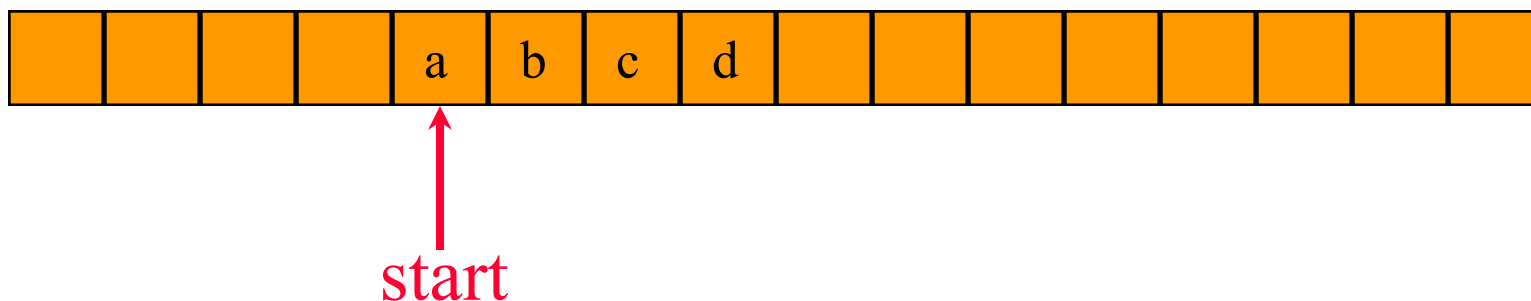
Ой



- Нэг хэмжээст массив  $x = [a, b, c, d]$
- Ойн дараалсан байрлалд тусна
- $\text{байршил}(x[i]) = \text{start} + i$

# НЭМЭЛТ ОЙН ХЭМЖЭЭ

Ой



$$\begin{aligned}\text{НЭМЭЛТ ой} &= 4 \text{ байт ( start )} \\ &\quad + 4 \text{ байт ( x.length )} \\ &= 8 \text{ байт}\end{aligned}$$

(**x** массивын элементүүдийг хадгалах ой ороогүй)

# Хоёр хэмжээст массив

Хоёр хэмжээст массив **a** –г зарлах:

```
int [][]a = new int[3][4];
```

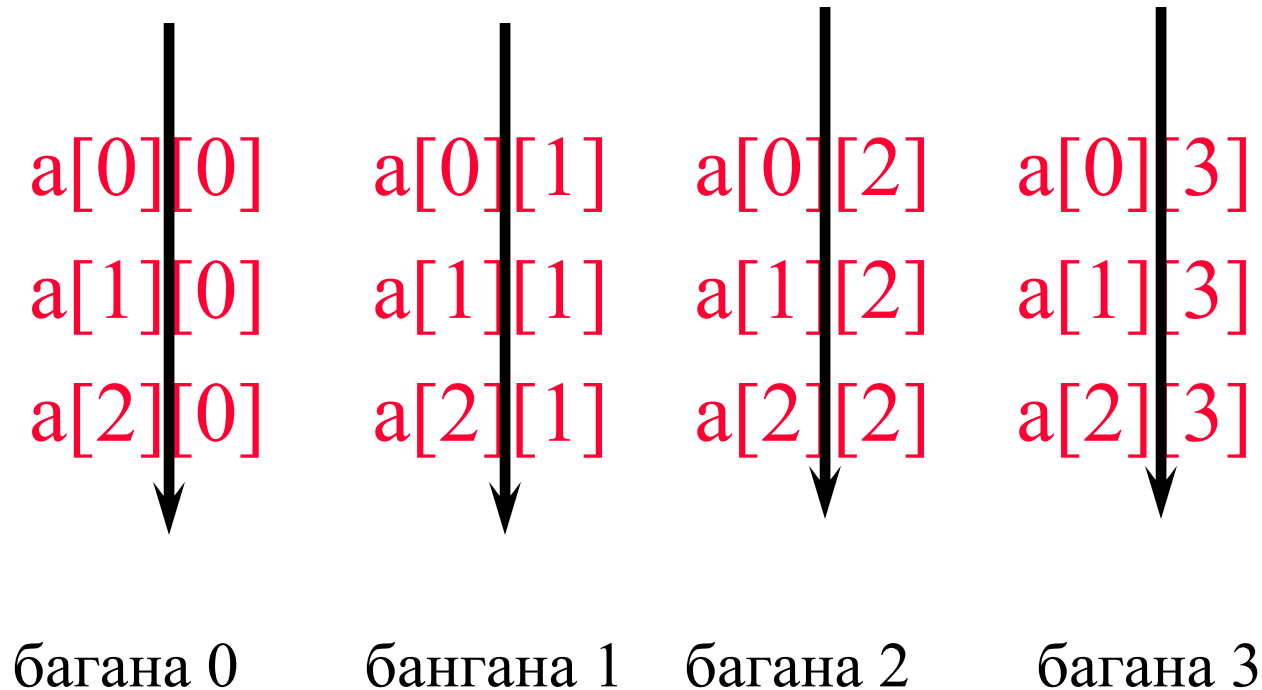
Гэж зарлагдсан массивын элементүүдийг  
хүснэгт байдлаар үзүүлбэл:

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

# Хоёр хэмжээст массивын мөрүүд

<del><math>a[0][0]</math></del>	<del><math>a[0][1]</math></del>	<del><math>a[0][2]</math></del>	<del><math>a[0][3]</math></del>	→ мөр 0
<del><math>a[1][0]</math></del>	<del><math>a[1][1]</math></del>	<del><math>a[1][2]</math></del>	<del><math>a[1][3]</math></del>	→ мөр 1
<del><math>a[2][0]</math></del>	<del><math>a[2][1]</math></del>	<del><math>a[2][2]</math></del>	<del><math>a[2][3]</math></del>	→ мөр 2

# Хоёр хэмжээст массивын баганууд



# Java, C, C++ -д хоёр хэмжээст массивыг дүрслэх

2-хэмжээст массив **x**

a, b, c, d

e, f, g, h

i, j, k, l

2D массивыг 1D мөрүүдийн массив

**x = [row0, row1, row2]**

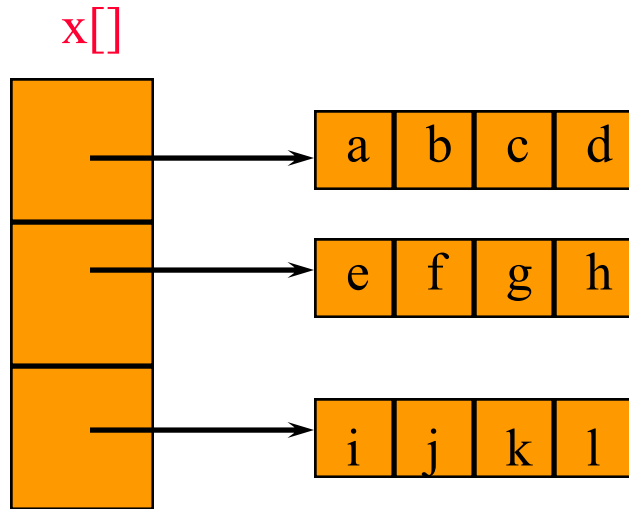
**row0 = [a,b, c, d]**

**row1 = [e, f, g, h]**

**row2 = [i, j, k, l]**

гэж **4** 1D массив байдлаар хадгалж болно

# Java, C, C++ -д хоёр хэмжээст массивыг дүрслэх

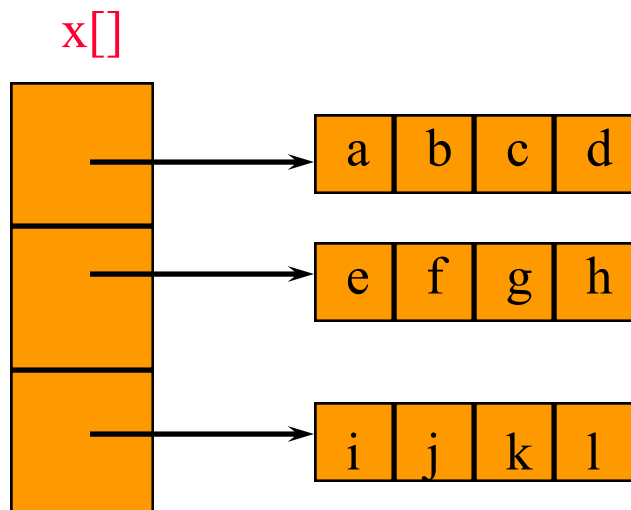


x.length = 3

x[0].length = x[1].length = x[2].length = 4

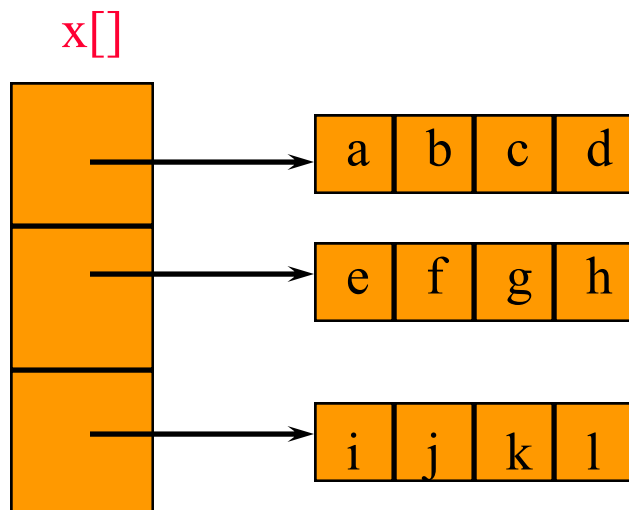


# НЭМЭЛТ ОЙН ХЭМЖЭЭ



НЭМЭЛТ ОЙ = 4 1D массивын нэмэлт ой  
=  $4 * 8$  байт  
= 32 байт  
= (мөрийн тоо + 1) x 8 байт

# Java, C, C++ -д массивыг дүрслэх



- Ийм дүрслэлийг **массивуудын-массив** дүрслэл гэнэ.
- 4 1D массивт 3, 4, 4, 4 гэсэн хэмжээстэй дараалсан ой шаардагдана.
- **number of rows** хэмжээтэй 1 блок + **number of columns** хэмжээтэй **number of rows** блокууд

# Мөрийг чухалчилсан тусгал

- 3 х 4 массивын жишээ:

a b c d

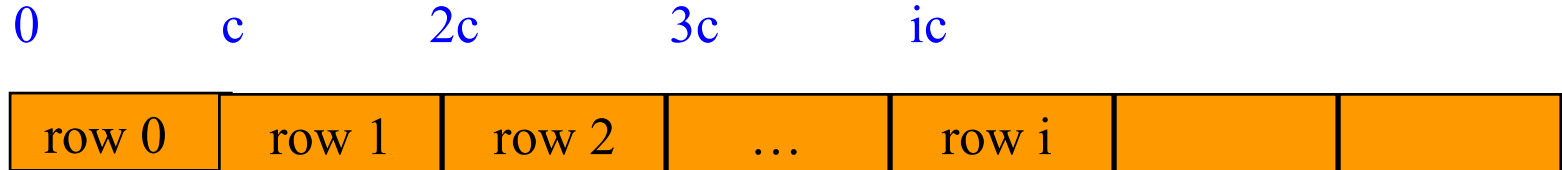
e f g h

i j k l

- Элементүүдийг мөр мөрөөр нь цуглуулж 1D  $y$  массивт хувиргах.
- Элемээнтүүд мөрөндөө зүүнээс баруун тийш.
- Мөрүүд дээрээс доош цуварна.
- Тэгвэл  $y[] = \{a, b, c, d, e, f, g, h, i, j, k, l\}$

row 0	row 1	row 2	...	row i		
-------	-------	-------	-----	-------	--	--

# Элемент $x[i][j]$ олох



- $x$  массив  $r$  мөр,  $c$  баганатай гэвэл
- Мөр бүр  $c$  элементтэй
- $i$  дэх мөрийн өмнө  $i$  мөр орно
- ө.х.  $ic$  элемент  $x[i][0]$  –ий өмнө орно
- ө.х.  $x[i][j]$  1D массивын  $ic + j$  баршилд хадгалагдана

# НЭМЭЛТ ОЙН ХЭМЖЭЭ

row 0	row 1	row 2	...	row i		
-------	-------	-------	-----	-------	--	--

$$\begin{aligned} & 4 \text{ байт ( } \textcolor{red}{start} \text{ - 1D массив )} + \\ & 4 \text{ байт ( } \textcolor{red}{length} \text{ - 1D массив )} + \\ & 4 \text{ байт ( } \textcolor{red}{c} \text{ (баганын тоо) } \\ & = 12 \text{ байт} \end{aligned}$$

$$(\text{мөрийн тоо} = \textcolor{red}{length} / \textcolor{red}{c})$$

# Сул тал

**rc** урттай дараалсан ой хэрэгтэй.

# Баганыг чухалчилсан тусгал

a b c d

e f g h

i j k l

- Элементүүдийг багана баганаар нь цуглуулж 1D **y** массивт хувиргах.
- Элементүүд баганадаа дээрээс доош.
- Баганууд зүүнээс баруун тийш цуварна.
- Тэгвэл **y** = {a, e, i, b, f, j, c, g, k, d, h, l}

# Матриц

Утгын хүснэгт. Мөр, баганатай, 0 биш 1-ээс эхэлж дугаарлагдана.

a b c d мөр 1

e f g h мөр 2

i j k l мөр 3

- $x[i][j]$  оронд  $x(i,j)$  тэмдэглэгээг хэрэглэе.
- Матрицыг 2D массиваар дүрсэлж болно.



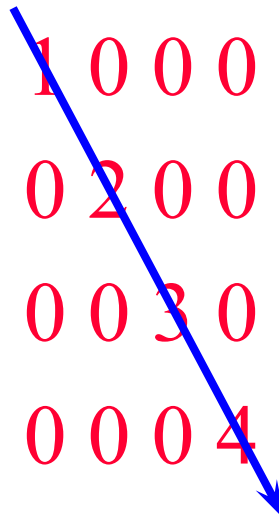
# Матрицад 2D массив хэрэглэхийн дутагдал

- Индекс 1 –ээр өөрчлөгддөг.
- Java -ийн массивт матрицын **add**, **transpose**, **multiply**, зэрэг үйлдлүүд байхгүй.
  - $x$ ,  $y$  2D массив бол.  $x + y$ ,  $x - y$ ,  $x * y$ , гэж болохгүй.
- Объект хандалтат **Matrix** классыг хөгжүүлснээр матрицын бүх үйлдлийг шийдэж болно. Сурах бичгээс хар.

# Диагональ матриц

$n \times n$  матрицын тэг биш бүх утгууд  
диагональ дээр байрлана.

# Диагональ матриц



1	0	0	0
0	2	0	0
0	0	3	0
0	0	0	4

- $x(i,j)$  диагональ дээр байхын тулд  $i = j$
- $n \times n$  диагналийн элементийн тоо  $n$
- Диагналийн бус элемент тэг байна
- $n^2$  элементийн оронд зөвхөн диагналийн элементийг хадгална

# Доод гурвалжин матриц

$n \times n$  матрицын тэг биш элементүүд диагнль болон түүнээс доош байрлана.

1 0 0 0

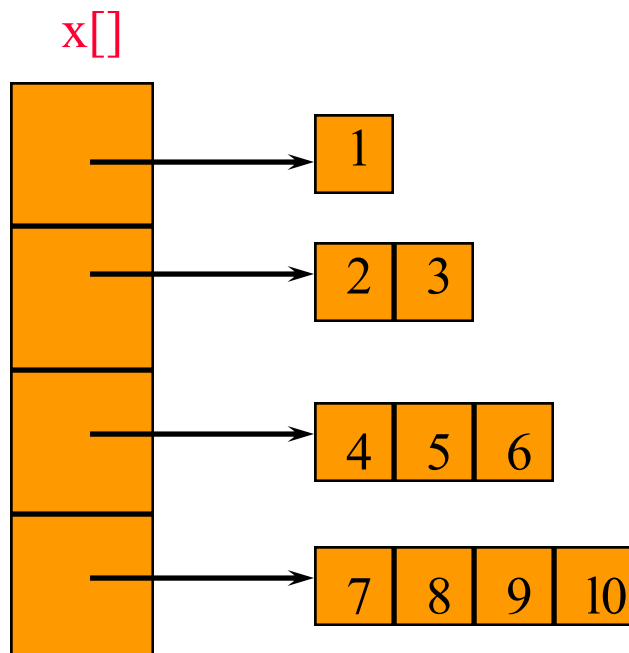
2 3 0 0

4 5 6 0

7 8 9 10

- $x(i,j)$  доод гурвалжинд байхын тулд  $i \geq j$ .
- Доод гурвалжингийн элементийн тоо  $1 + 2 + \dots + n = n(n+1)/2$ .
- Зөвхөн доод гурвалжинг хадгална

# Массивуудын массив дүрслэл



Ердийн биш 2D массив ... мөрүүд дэх  
элементийн тоо харилцан адилгүй байна.

# Ердийн биш массивыг үүсгэж, ашиглах

// хоёр хэмжээст массивын хувьсагчийг зарлахдаа

// мөрийн тоог зааж өгөх

```
int [][] irregularArray = new int [numberOfRows][];
```

// мөр бүрийн элементэд зай хуваарилах

```
for (int i = 0; i < numberOfRows; i++)
```

```
    irregularArray[i] = new int [size[i]];
```

// массивыг ердийн массивын адил ашиглах

```
irregularArray[2][3] = 5;
```

```
irregularArray[4][6] = irregularArray[2][3] + 2;
```

```
irregularArray[1][1] += 3;
```

# Доод гурвалжин массивыг 1D массиваар байршуулах

Мөрийг чухалчлах дарааллийг ашиглахдаа  
доод гурвалжинд орохгүй элементийг  
орхино.

Матрицын жишээ

1 0 0 0

2 3 0 0

4 5 6 0

7 8 9 10

бол

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

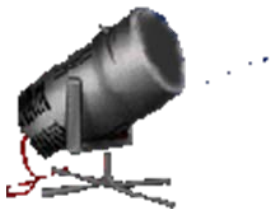
## $[i][j]$ элементийн индекс

0	1	3	6			
r 1	r2	r3	...	row i		

- Дараалал: мөр 1, мөр 2, мөр 3, ...
- Мөр  $i$  дараах мөрүүдийн өмнө 1, 2, ...,  $i-1$
- $i$  –р мөрийн хэмжээ  $i$ .
- $i$  -р мөрөөс өмнө орсон элементийн тоо  
 $1 + 2 + 3 + \dots + i-1 = i(i-1)/2$
- Иймд элемент  $(i,j)$  1D массивын  $i(i-1)/2 + j - 1$  байршилд байна.



# Сийрэг матриц



сийрэг ... олон элемент нь тэг  
нягт ... цөөн элемент тэг

# Сийрэг матрицын жишээ

диагналь

гурвалсан диагналь

доод гурвалжин (?)

Эд бүтцийн сийрэг матриц.

1D массиваар илэрхийлж болно, элементийг олохдоо тусгалын функц ашиглана.

# Бүтцийн бус сийрэг матриц

Нислэгийн матриц.

- Нисэх буудлын дугаар  $1 - n$
- $\text{flight}(i,j)$  = list of nonstop flights from airport  $i$  буудлаас  $j$  буудал хүртэл зогсолтгүй нисэх нислэгийн жагсаалт
- $n = 1000$  (гэж үзье)
- $n \times n$  массивын байж болох жагсаалт  $\Rightarrow 4$  сая байт
- Нийт нислэгийн тоо =  $20,000$  (гэж үзье)
- $20,000$  нислэгийн жагсаалтанд  $\Rightarrow 80,000$  байт хэрэгтэй болно

# Бүтцийн бус сийрэг матриц

Веб хуудасны матриц.

Веб хуудсууд  $1 - n$

$web(i,j) = i$  -ээс  $j$  хуудас хүрэх холбоос

Веб-ийн шинжилгээ.

# Веб хуудасны матриц

- $n = 2 \text{ billion}$  (өдөрт  $1 \text{ million}$  өсч байна)
- $n \times n$  бүхэл массив  $\Rightarrow 16 * 10^{18} \text{ bytes}$  ( $16 * 10^9 \text{ GB}$ )
- Хуудас бүр  $10$  (жишээ нь) хуудастай дунджаар холбогддог
- Дунджаар нэг мөрөнд  $10$  тэг биш оролт
- Тэг биш элементийг хадгалах шаардагдах ой ойролцоогоор  $20 \text{ billion} \times 4 \text{ bytes} = 80 \text{ billion bytes}$  ( $80 \text{ GB}$ )

# Бүтцийн бус сийрэг матрицын дүрслэл

Мөрийг чухалчилсан дан шугаман жагсаалт.

мөрийг чухалчлах дарааллаар сийрэг матрицын тэг  
биш элементийг шүүрдэх

тэг биш элемент бүрийг дараах байдлаар дүрслэх

(row, column, value)

энэ гурвалсан жагсаалтыг массив жагсаалт, эсхүл  
холбоост-гинжин жагсаалтаар дүрсэлнэ

# Дан шугаман жагсаалтын жишээ

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0

list =

row	1	1	2	2	4	4
column	3	5	3	4	2	3
value	3	4	5	7	2	6

# Массив шугаман жагсаалтын дүрслэл

list =

row	1	1	2	2	4	4
column	3	5	3	4	2	3
value	3	4	5	7	2	6

element	0	1	2	3	4	5
row	1	1	2	2	4	4
column	3	5	3	4	2	3
value	3	4	5	7	2	6



# Гинжин дүрслэл

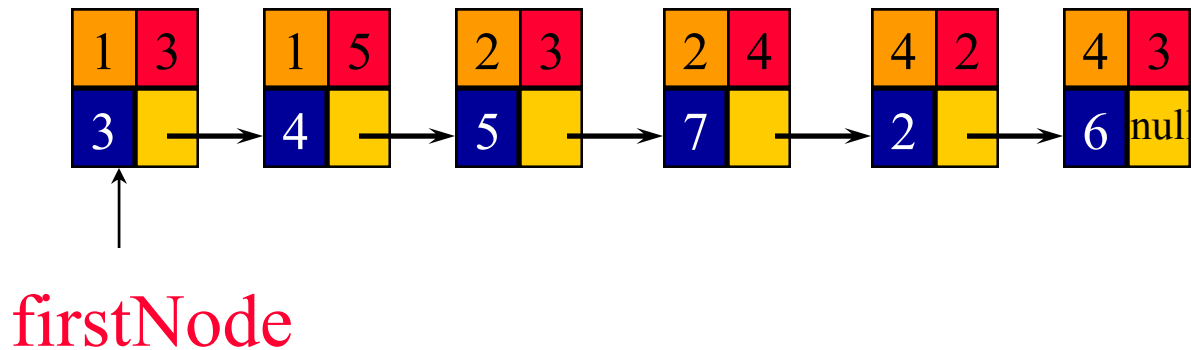
Зангилааны бүтэц.

row	col
value	next

# ЭНГИЙН ГИНЖ

list = 

row	1	1	2	2	4	4
column	3	5	3	4	2	3
value	3	4	5	7	2	6



# Мөр бүрт нэг шугаман жагсаалт

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0

row1 = [(3, 3), (5,4)]

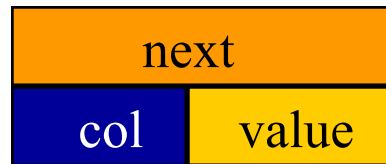
row2 = [(3,5), (4,7)]

row3 = []

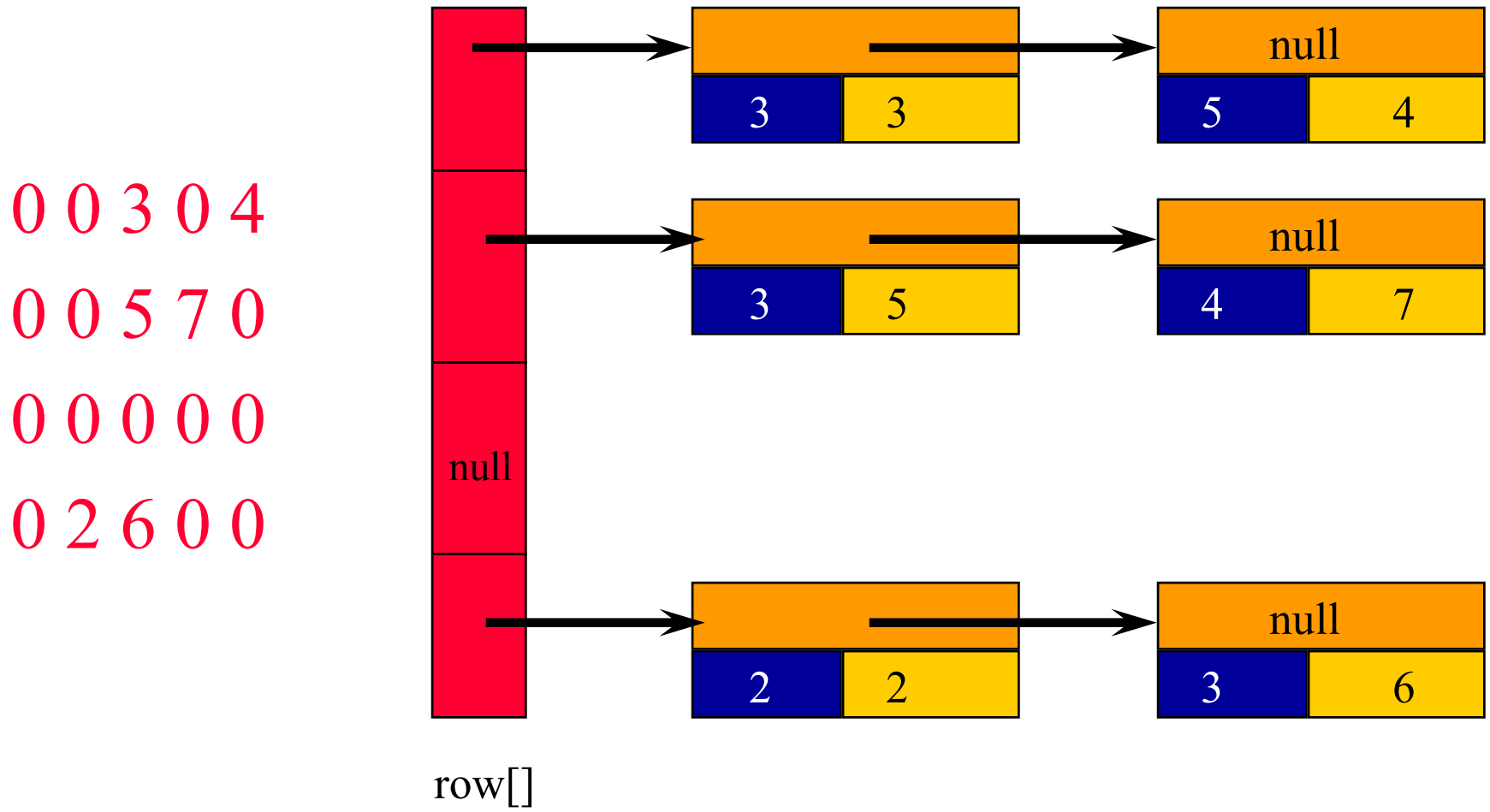
row4 = [(2,2), (3,6)]

# Мөрний массивын гинж

Зангилааны бүтэц.



# Мөрний массивын гинж



# Ортогналь жагсаалтын дүрслэл

Мөр ба баганын жагсаалт.

Зангилааны бүтэц.

row	col	value
down	next	

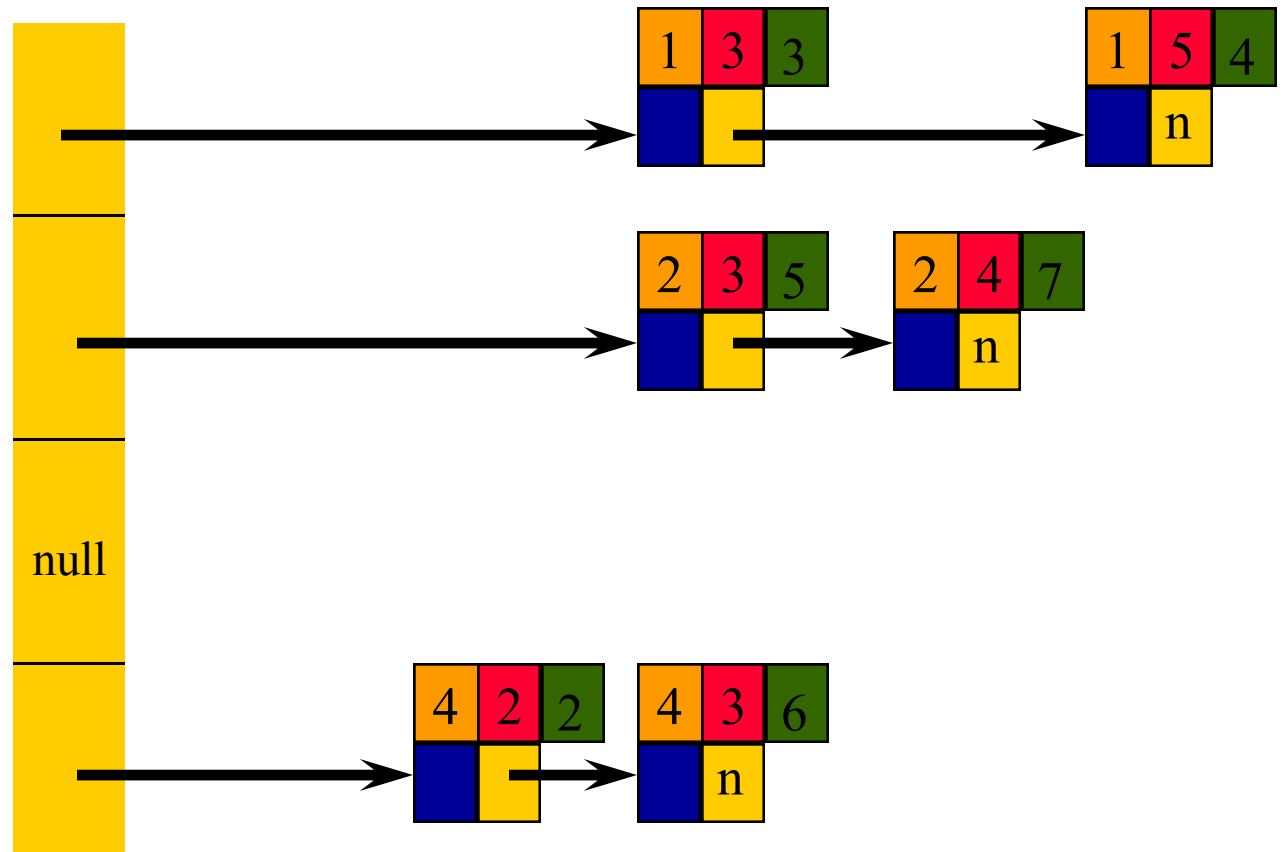
# Мөрний жагсаалт

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



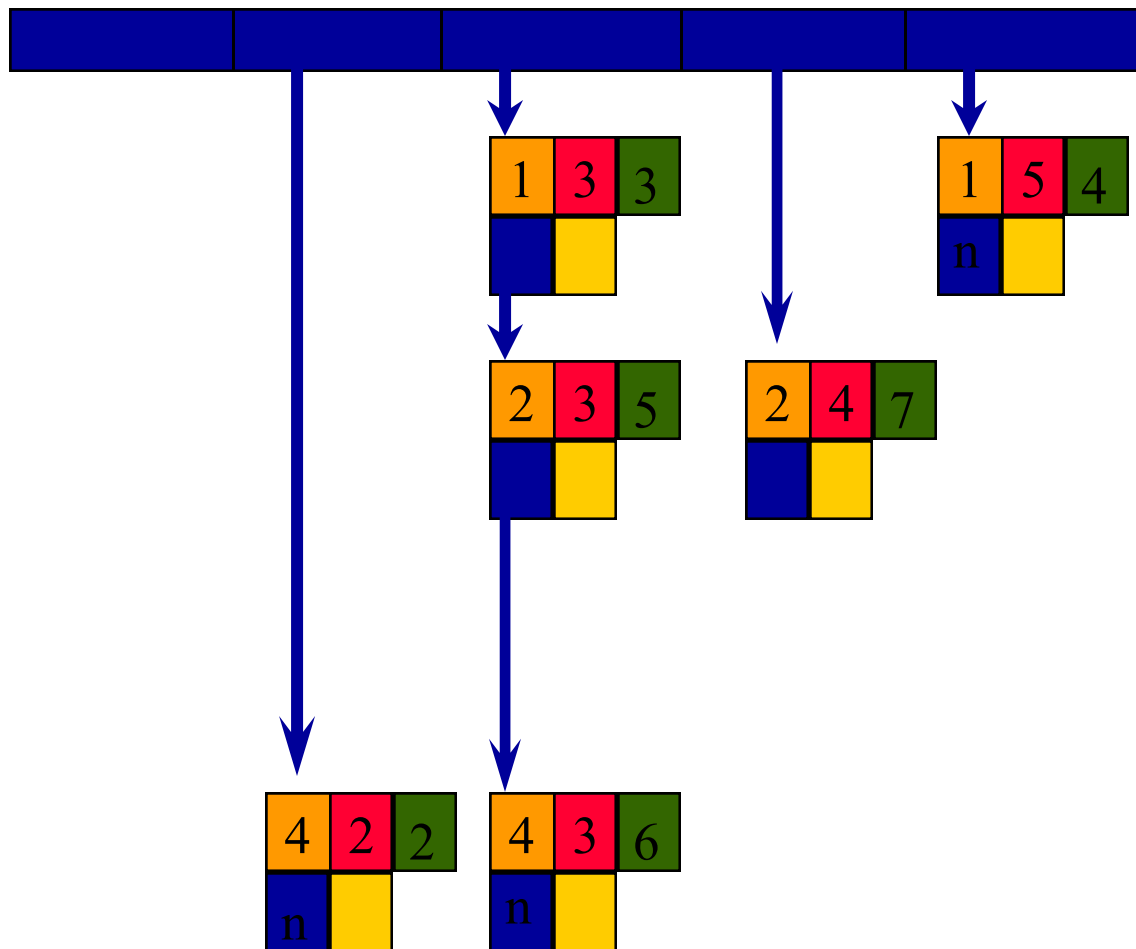
# Баганын жагсаалт

0 0 3 0 4

0 0 5 7 0

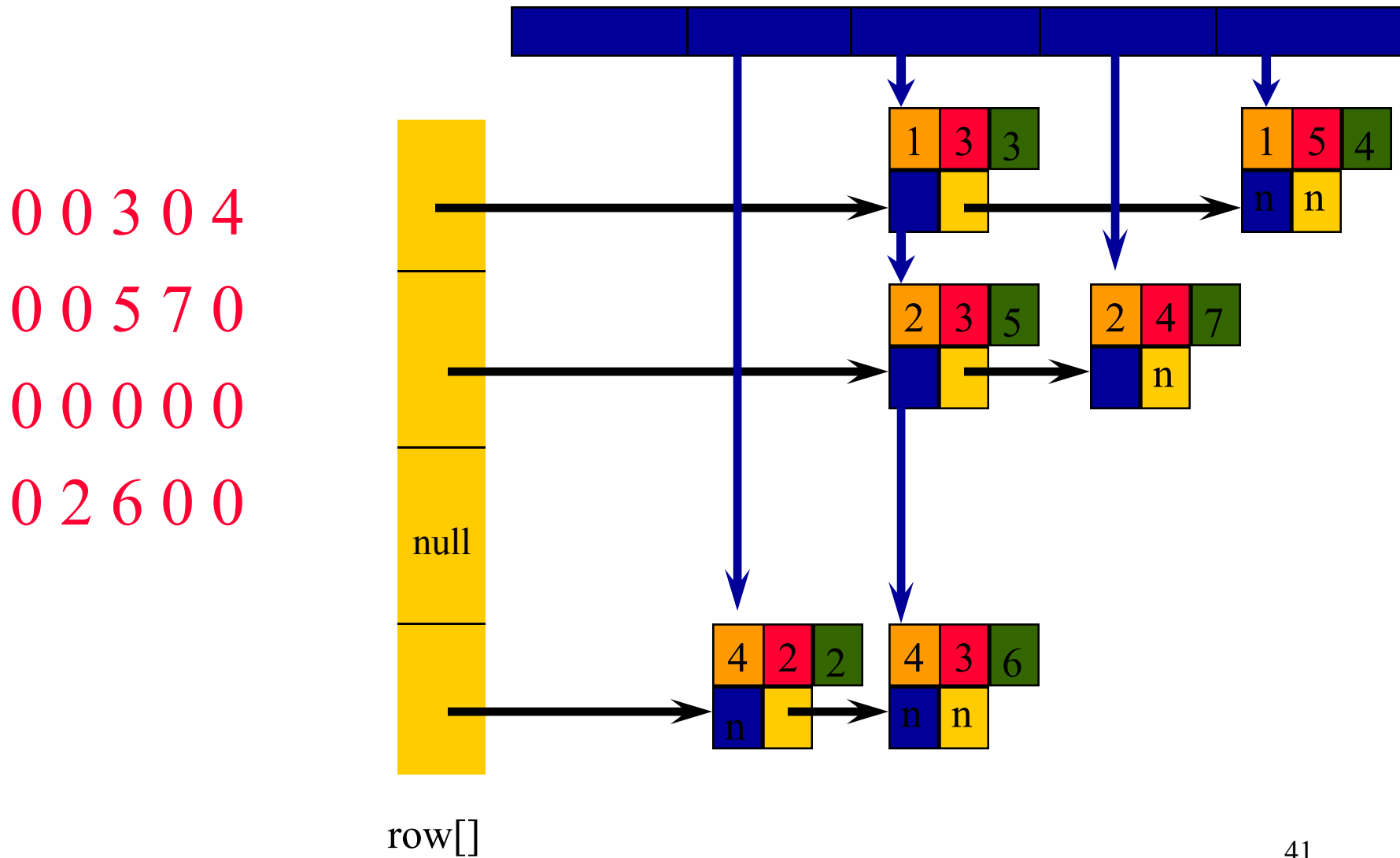
0 0 0 0 0

0 2 6 0 0





# Ортогналь жагсаалт



# Хувилбарууд

Гинжны оронд цагираг жагсаалтыг  
ашиглаж болно.

# Ойн ойролцоо хэрэгцээ

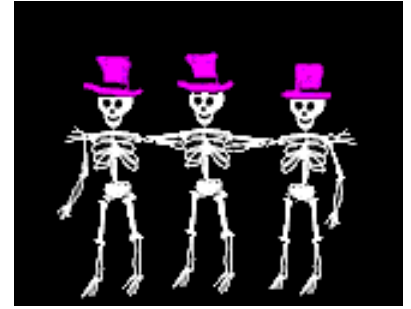
500 x 500 матриц, 1994 тэг биш элемент

2D массив  $500 \times 500 \times 4 = 1\text{million}$  байт

Дан массив жагсаалт  $3 \times 1994 \times 4 = 23,928$  байт

Мөр бүрт 1 гинж  $23928 + 500 \times 4 = 25,928$

# Хугацааны үзүүлэлт



Matrix Transpose

500 x 500 матриц, 1994 тэг биш элемент

2D массив

210 ms

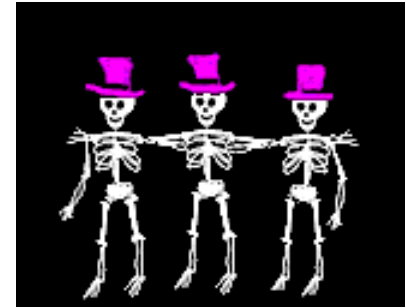
Дан массив жагсаалт

6 ms

Мөр бүрт 1 гинж

12 ms

# Хугацааны үзүүлэлт



Matrix Addition.

500 x 500 матриц, 1994 ба 999 тэг биш  
элемент

2D массив

880 ms

Дан массив жагсаалт

18 ms

Мөр бүрт 1 гинж

29 ms