# Sparse Matrices

Dr. Sambit Bakshi
CSE

# Sparse Matrices

sparse … many elements are zero

dense … few elements are zero

# Example of Sparse Matrices

- diagonal
- tridiagonal
- lower triangular
- upper triangular

These are structured sparse matrices.

$$A = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 13 \end{bmatrix}$$

**Diagonal matrix**

$$B = \begin{bmatrix} 9 & 5 & 0 & 0 \\ 4 & 7 & 2 & 0 \\ 0 & 3 & 5 & 2 \\ 0 & 0 & 1 & 13 \end{bmatrix}$$

**Tri-diagonal matrix**

$$C = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 4 & 7 & 0 & 0 \\ 5 & 3 & 5 & 0 \\ 2 & 6 & 1 & 13 \end{bmatrix}$$

**Lower triangular matrix**

$$D = \begin{bmatrix} 9 & 5 & 9 & 1 \\ 0 & 7 & 2 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 13 \end{bmatrix}$$

**Upper triangular matrix**

# Unstructured Sparse Matrices

Airline flight matrix.

- airports are numbered 1 through n
- flight(i,j) = no of nonstop flights from airport i to airport j
- n = 1000 (say)
- n x n array of list references => 2 million bytes
- total number of flights = 20,000 (say)
- need at most 20,000 references => at most 40,000 bytes

# Unstructured Sparse Matrices

Web page matrix.

web pages are numbered 1 through $n$

$web(i,j)$ = number of links from page $i$ to page $j$

# Web Page Matrix

- n = 2 billion (and growing by 1 million a day)
- n x n array of ints => $8 * 10^{18}$ bytes (≈ $8 * 10^9$ GB)
- each page links to 10 (say) other pages on average
- on average there are 10 nonzero entries per row
- space needed for nonzero elements is approximately 20 billion x 2 bytes = 40 billion bytes (40 GB)

# Representation Of Unstructured Sparse Matrices

- **Triplet Array**
- **Chain representation**
  - **Single Chain**
  - **Row Chain**
  - **Column Chain**
  - **Orthogonal Chain**

# Representation Of Unstructured Sparse Matrices using triplet array

$$A = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

| **4** | **5** | **6** |
|---|---|---|
| 1 | 3 | 3 |
| 1 | 5 | 4 |
| 2 | 3 | 5 |
| 2 | 4 | 7 |
| 4 | 2 | 2 |
| 4 | 3 | 6 |

(row, column, value) --- triplet

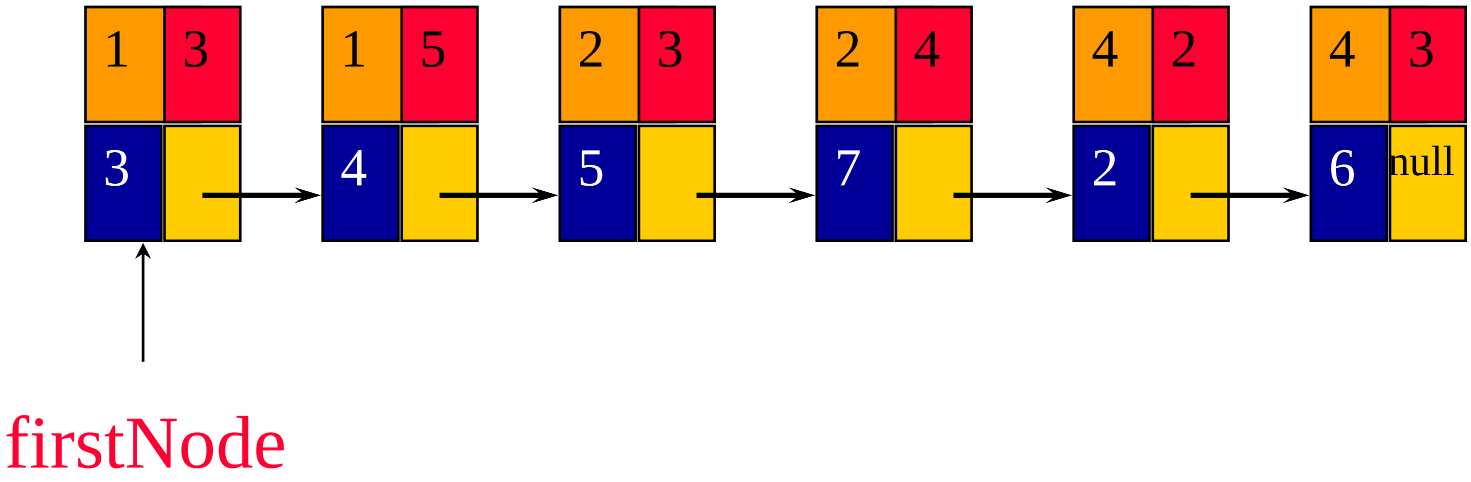
# Chain Representation

Node structure.

| | |
|---|---|
| row | col |
| value | next |

# Single Chain

$$A = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$



# One Linear List Per Row

$$A = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

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

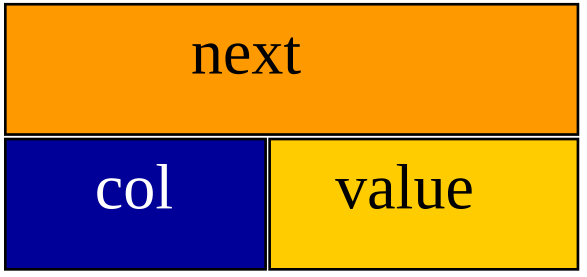
row2 = [(3,5), (4,7)]

row3 = []

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

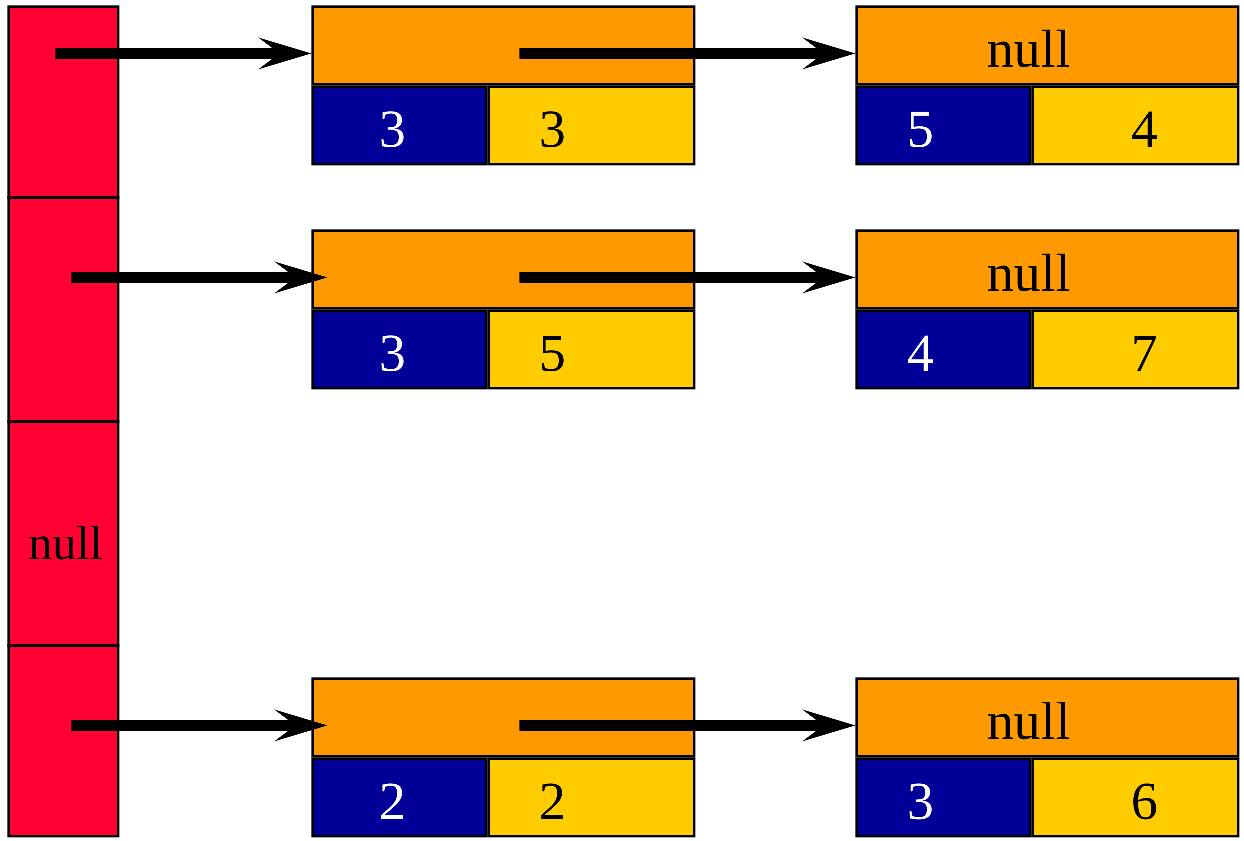
# Array Of Row Chains

Node structure.



# Array Of Row Chains

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 3 | 0 | 4 |
| 0 | 0 | 5 | 7 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 6 | 0 | 0 |




*row[]

# One Linear List Per Column

$$A = \begin{bmatrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{bmatrix}$$

col1 = []

col2 = [(4,2)]

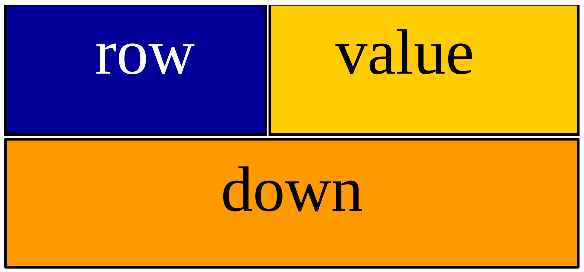
col3 = [(1, 3), (2,5), (4,6)]

col4 = [(2,7)]

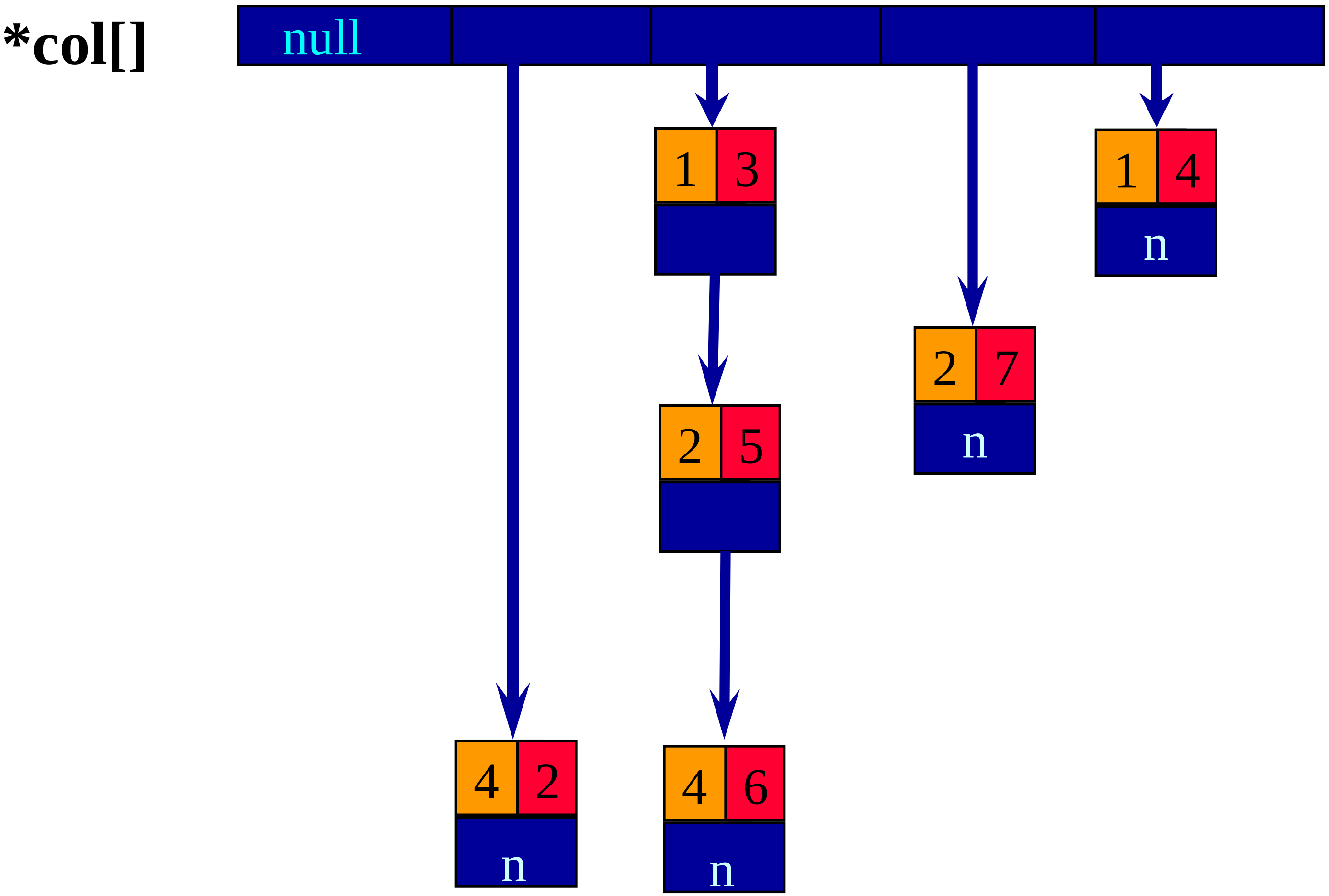
col5 = [(1,4)]

# Array Of Column Chains

Node structure.



# Array Of Column Chains

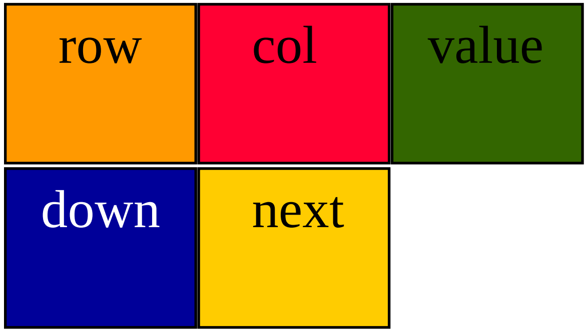



| | | | | |
|---|---|---|---|---|
| 0 | 0 | 3 | 0 | 4 |
| 0 | 0 | 5 | 7 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 6 | 0 | 0 |

# Orthogonal List Representation

Both row and column lists.

Node structure.



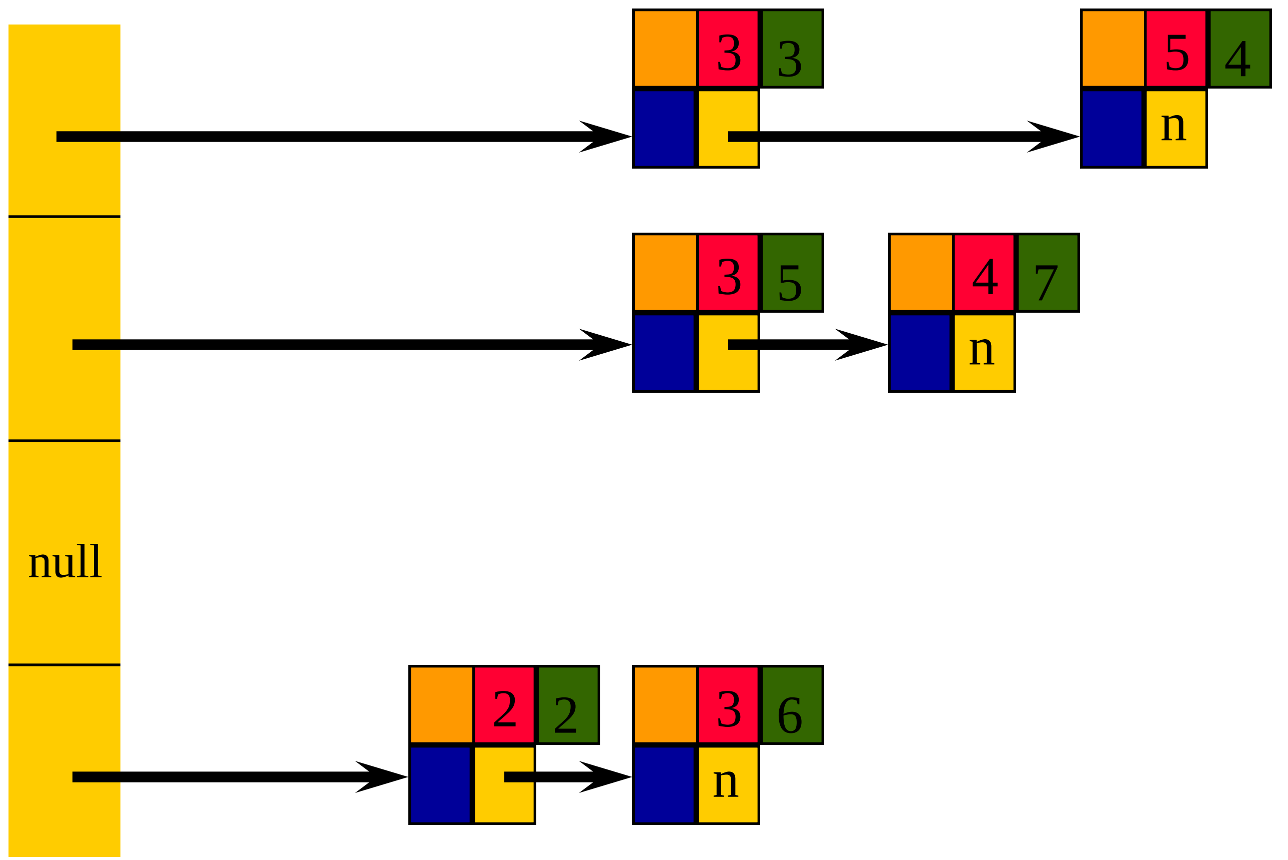
# Row Lists

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



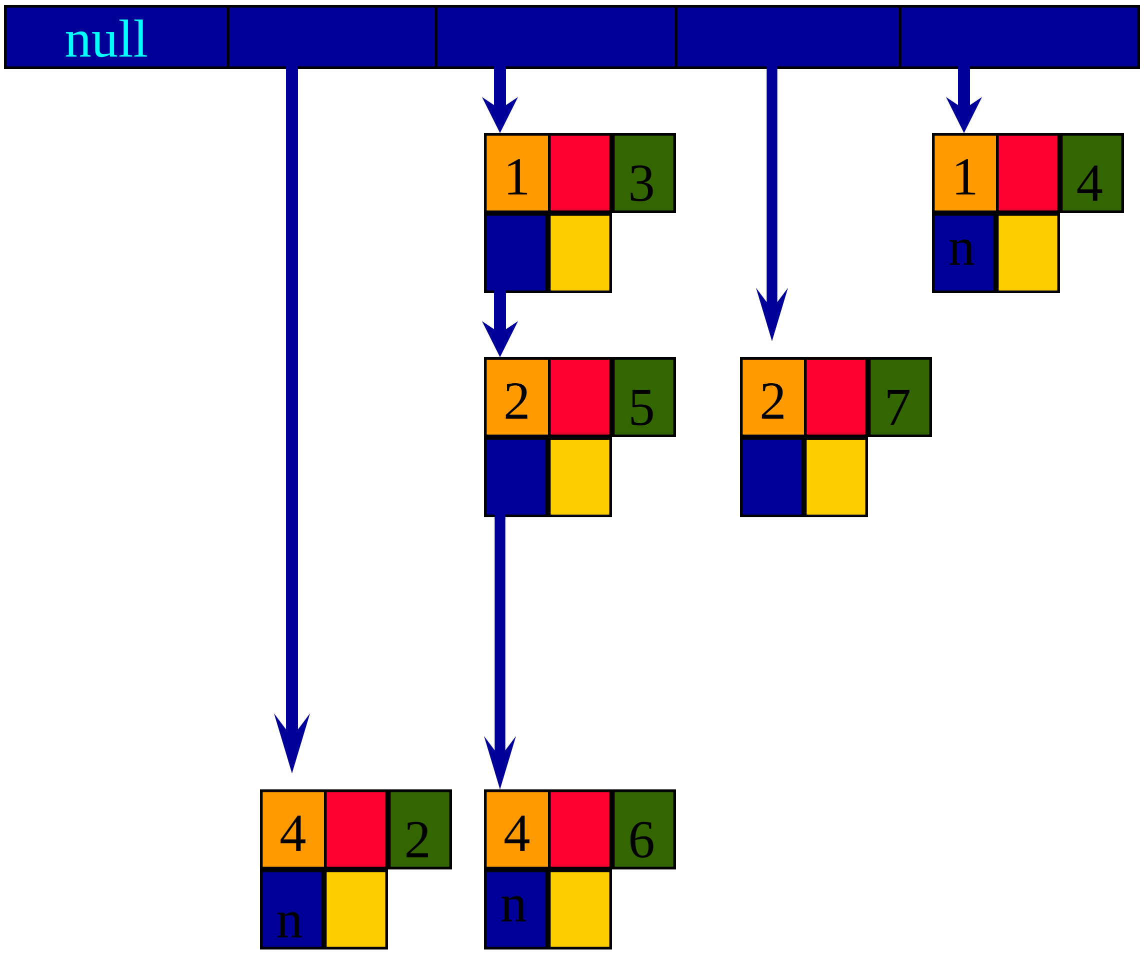
# Column Lists

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



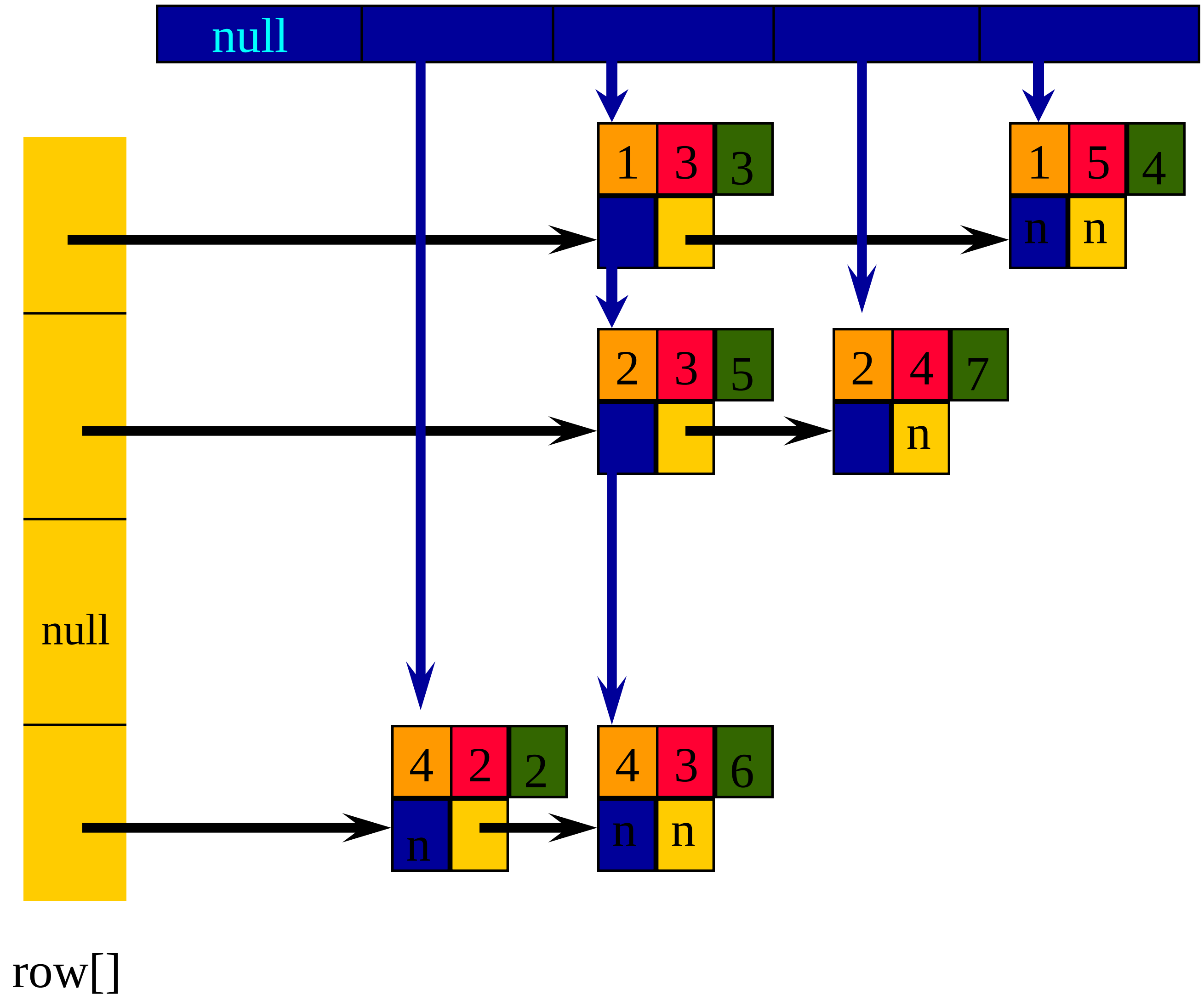
# Orthogonal Lists

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



# Storage Requirements

500 x 600 integer matrix with 1994 nonzero elements

| | |
|---|---|
| 2D array | 500 x 500 x 2 = 0.5million bytes |
| Triplet array | 3 x 1994 x 2 = 11,970 bytes |
| Single chain | 1994 x 8 (node size) + 2 (head pointer) = 15954 bytes |
| One Chain Per Row | 1994 x 6 (node size) + 2 x 500 (number of rows) = 12,964 bytes |

**how many bytes for column chain and orthogonal chain?**