Lists. Variations

- Multiple Values Per Node

ULNode = record

next: Position

elemCount: integer

elemData: array[0..MAX-1] of TElement

end

Position = ^**ULNode**

Terminology: **unrolled linked list**

Lists. Variations

- with multiple links from nodes

Node = ***record:***

info: **TElement**

links: List<Position> ***//array, record, linked***

end

Terminology: General Linked Lists

Multiply Linked List

Multi-Linked Lists

Multidimensional arrays

For example:

C++ : described as "arrays of arrays".

```
int matrix [3][5];
```

```
matrix[1][3]
```

the second element vertically and fourth horizontally

Using STL : Vector based multi-dim. array

```
vector<vector<double> > array2D
// Put some values in
array2D[1][2] = 6.0
```

Jagged arrays

many rows of varying lengths

For example:

```
int jagged_row0[] = {0,1};
int jagged_row1[] = {1,2,3};
int *jagged[] = { jagged_row0,
                  jagged_row1 };
```

Lists. Variations. Application

Sparse Matrices

Sparse Matrix

- sparse … many elements are zero
- dense … few elements are zero

Structured sparse

- diagonal
- tridiagonal
- upper/lower triangular (?)

Unstructured sparse matrix. Example

- Web page matrix.
  - web pages are numbered 1 through n
  - web(i,j) = number of links from page i to page j
- n ~ 10^9
- n x n array => 10^{18} consecutive position
  (linear representation)
- each page links to 10 (say) other pages on average
  on average there are 10 nonzero entries per row
- space needed for non-*empty* elements is approximately
  1 billion x 10 = 10^{10} consecutive elements

- **sparse matrix** is a matrix populated primarily with zeros
- How to store it will depend at least partly on exactly how sparse it is, and what you want to do.
  - For some applications, just treating it as a regular matrix is just fine (especially if the dimensions aren't very big). A 15x15 sparse matrix isn't a big memory hog.
  - suppose the sparse matrix has 10240x10240 elements, and you're using 8-byte floating point numbers: how much memory do you need?
- **Representation of sparse matrix (idea)**
  - list: keep information about non-zero cells
  - links: for fast access from one non-zero cell to the next, on the same raw/column ← general linked list

sparse matrix

Unstructured Sparse Matrices. Representations

linear list in row-major order.

- nonzero elements of the sparse matrix
- each nonzero element is represented by a triple

  (row, column, value)

➔the list of triples (linked , …)

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

One Linear List Per Row

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)]

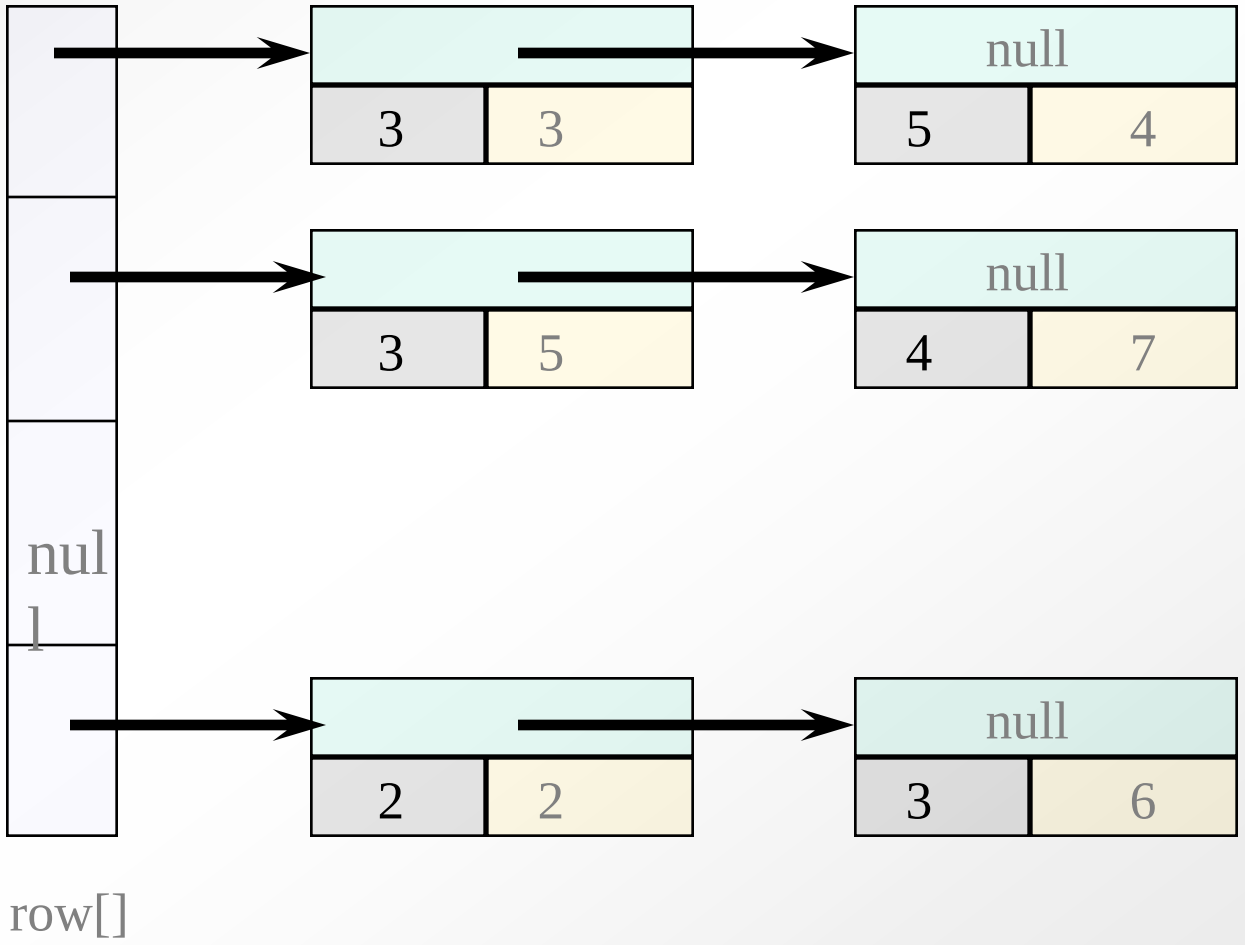
One Linear List Per Row (Linked)

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



Similar: one list per column

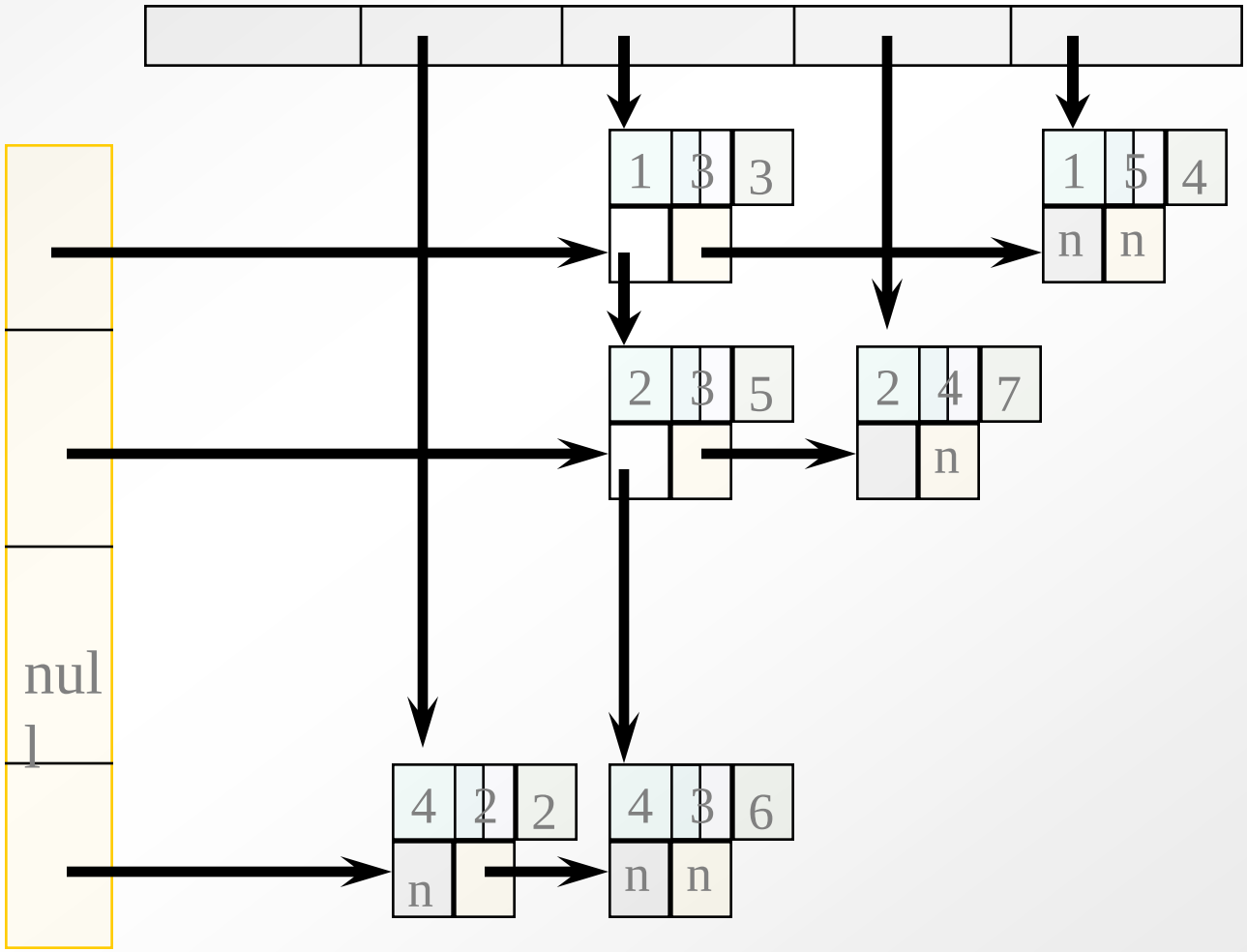
Orthogonal Lists

0 0 3 0 4

0 0 5 7 0

0 0 0 0 0

0 2 6 0 0



Variation: use circular lists