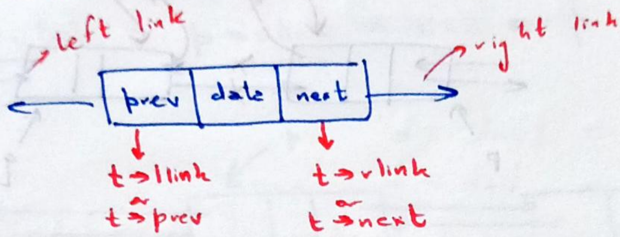
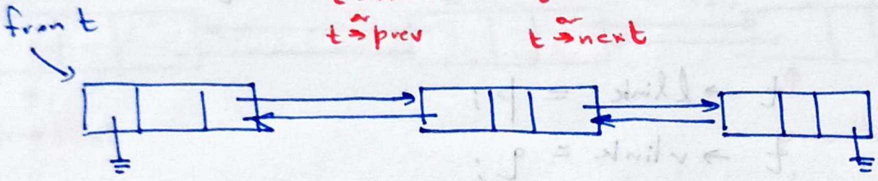
Double Linked List

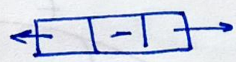


Eg

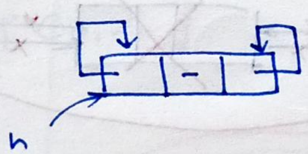


Circular Double Linked List

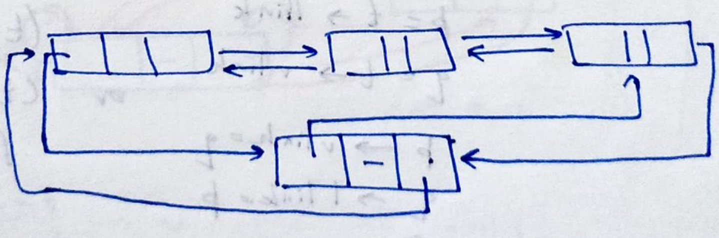
→ we use another node, which is called "head node", it's data is "don't care"



Empty list :



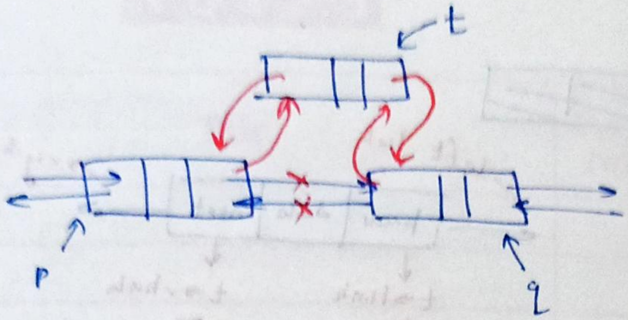
not empty list :



→ head node are used to simplify algorithms

Insertion

Eg



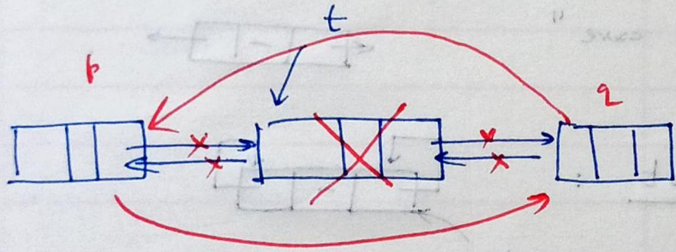
t → llink = p;

t → rlink = q;

p → rlink = t;

q → llink = t;

Delete



p = t → llink

q = t → rlink

p → rlink = q

q → llink = p

free (t)

or

(t → llink) → rlink = (t → rlink)

(t → rlink) → llink = t → llink

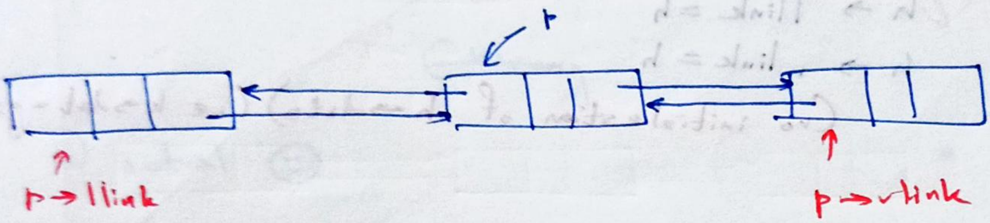
free (t)

~~while~~ while deleting, it may be case that the ~~remaining node list~~ list is already ~~becomes~~ ~~is~~ empty, i.e only head node remains., you can't delete head node (use if to ~~che~~ check first)

properties

for a node p:

$p \to llink \to rlink = p \to rlink \to llink = p$

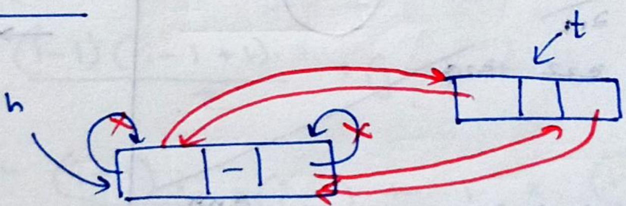


Summary

→ a node should have three things.

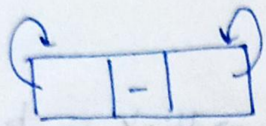
i) llink
ii) rlink
iii) Data

Insertion in empty list [Circular]



$t \to llink = h$
$t \to rlink = h$
$h \to rlink = t$
$h \to llink = t$

initial config of double ~~tm~~ circular link list



$h \rightarrow llink = h$

$h \rightarrow rlink = h$

(no initialization of $h \rightarrow data$) (i.e $h \rightarrow data$ = garbage)

Sparse Matrix in Linked Form

~~Reviews~~ Revision of Sparse Matrix representation in array form

I) If we have a matrix like this.

$$n \times n \quad \begin{bmatrix} a_{00} & & & & \\ a_{10} & a_{11} & & & \\ a_{21} & a_{22} & a_{22} & & \\ \vdots & & & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \rightarrow \text{lower triangular matrix.}$$

we can simplify the algo ~~to~~ for these special matrix.

→ no. of elements in the array = $1 + 2 + 3 + \cdots n = \frac{n(n+1)}{2}$

→ we store element in format

$a_{00}, a_{10}, a_{11}, a_{20}, a_{21}, a_{22} \cdots\cdots$

so we don't even need position array.

→ finding pos. Q) $B[L] = a_{ij}$, find L

i) possible ~~algo~~ :

$L = \cancel{i(1+} \; 1+2+3+\ldots\ldots i \cancel{-1} + j$ (i-1) wrong due to zero index matrix)

$= \cancel{\frac{i(i+1)}{2} + j}$ (wrong)

(~~Wrong~~

ii) actual ☺

$B[1] = a_{11}$
$B[2] = a_{12}$
$B[3] = a_{22}$
$B[4] = a_{31}$

→ assuming array is 1 indexed

$L = \underbrace{(1+2+\ldots+(i-1))}_{\text{top of } a_{ij}} + \underbrace{(j-1)}_{\text{Left of } a_{ij}} + \underbrace{1}_{\text{of actual } a_{ij}}$

~~index~~

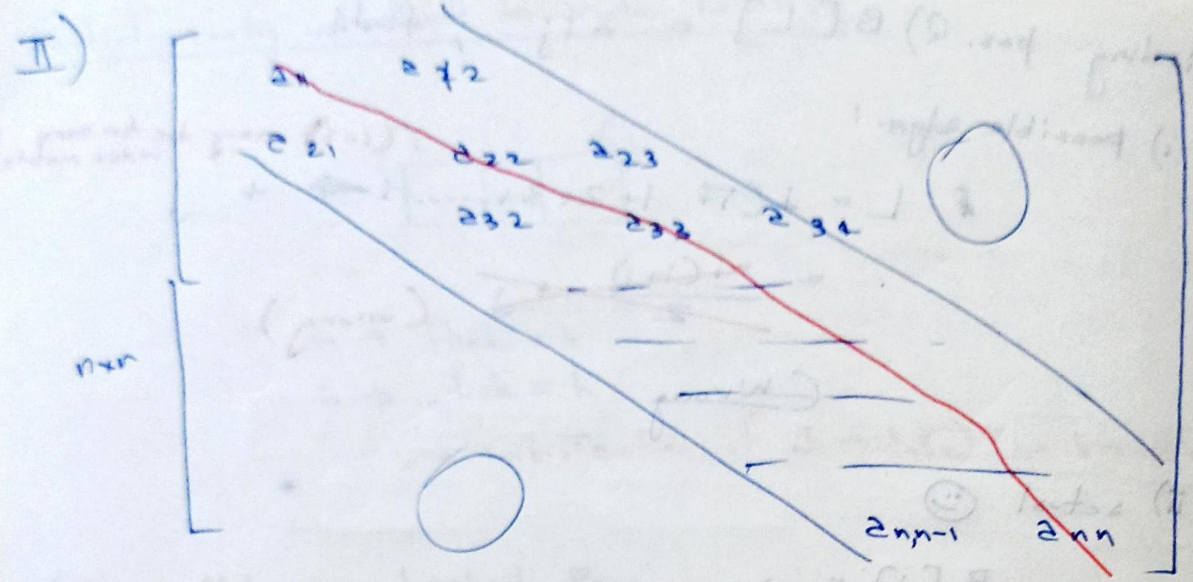
$= \frac{(i-1)(i-1+1)}{2} + (j-1) + 1$

$L = \frac{(i-1)\,i}{2} + (j-1) + 1 = \frac{(i-1)\,i}{2} + j$

∴ address ~~index~~ of $a_{ij} = \alpha + \left[\frac{i(i-1)}{2} + j \cancel{-1}\right] \times \underbrace{W}_{\text{sizeof(element)}}$

~~index of a_{ij} =~~

II)



→ main diagonal, it's upward and lower diaganol are non zero all other zero

→ called Tridiagonal matrix.

$B[1] = a_{11}$
$B[2] = a_{12}$
$B[3] = a_{21}$
$\vdots$

a) $B[L] = a_{ij}$ $L = ?$

$$\cancel{L = \underbrace{\overbrace{[\,2 + 3(i-2)\,]}^{\text{rows}}}_{\text{first row} \quad \text{next rows}} + \overbrace{(j-1)\%3}^{\text{columns}} + \overbrace{1}^{\text{itself}}}$$

element above a_{ij} = $2 + 3(i-2)$
element ~~above~~ left of a_{ij} = $j - i + 1$

$\therefore$ index of a_{ij} in B

$$L = [3(i-2) + 2] + [(j - i + 1)] + 1$$

$$= \underline{3i} - 6 + 2 + \underline{j} - \underline{i} + 1 + 1$$

$$L = 2i + j - 3 + 1$$

$$L = 2i + j - 2$$