

Always:

location of  $A[10]$



address

location of  $A[9][12]$  is a 2 d array.

In ~~all the~~ all subjects, all question we solve  
will be gate level or above gate level.

`int a[10] = {10, 20, 30, 40, 50, 60, ..... 100}`

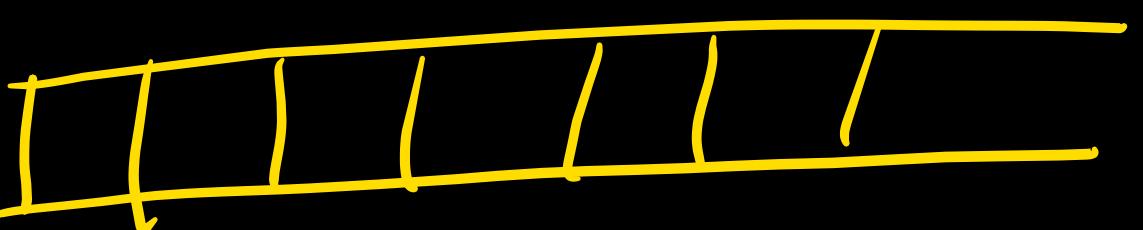
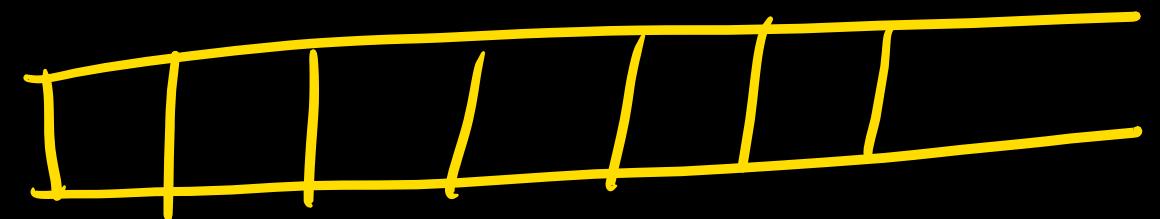


This array need not be ~~in~~ any programming language.

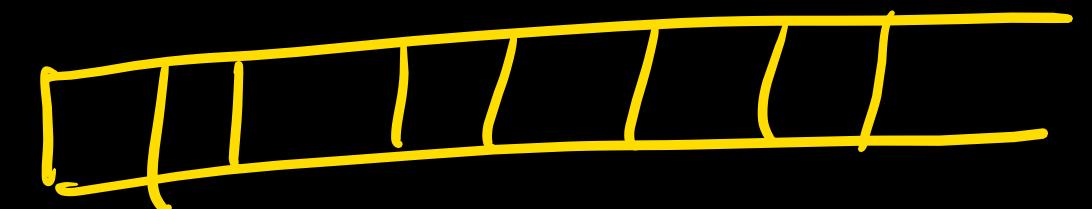
In programming languages we start with zero index.

In algo we start with 1 index in general

A starting index will be given ~~in~~ in question



,



100

given in the question

In C program :

int a[] = { 10, 20, 30.. - ... 100 }

↓  
no need to  
specify no size.

int a[]; → ~~wrong~~ wrong.

↓  
size is  
required.

1000	02	04	06	08	10	12	14	16	18
1	2	3	4	5	6	7	8	9	10
10	20	30	40	50	60	70	80	90	100

Each cell is 2 bytes

First item add = 1000

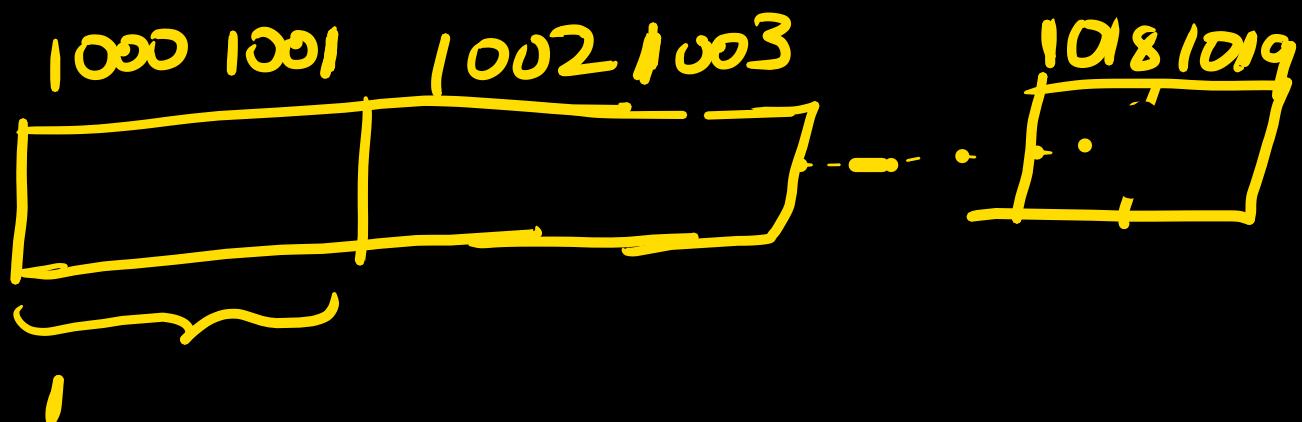
~~first~~ I address = 1000

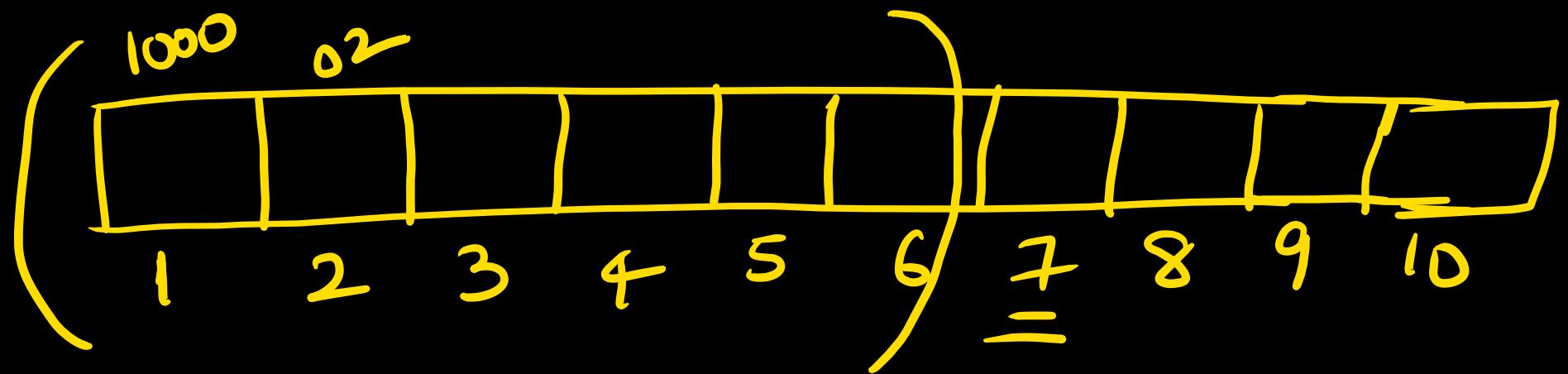
Last address = 1019

$$\text{total addresses} = 1019 - 1000 + 1$$

$$= 20 \text{ bytes}$$

$$\begin{aligned}\text{Total elements} &= 10 - 1 + 1 \\ &= 10 \text{ elements}\end{aligned}$$





Size of each element = 2 bytes

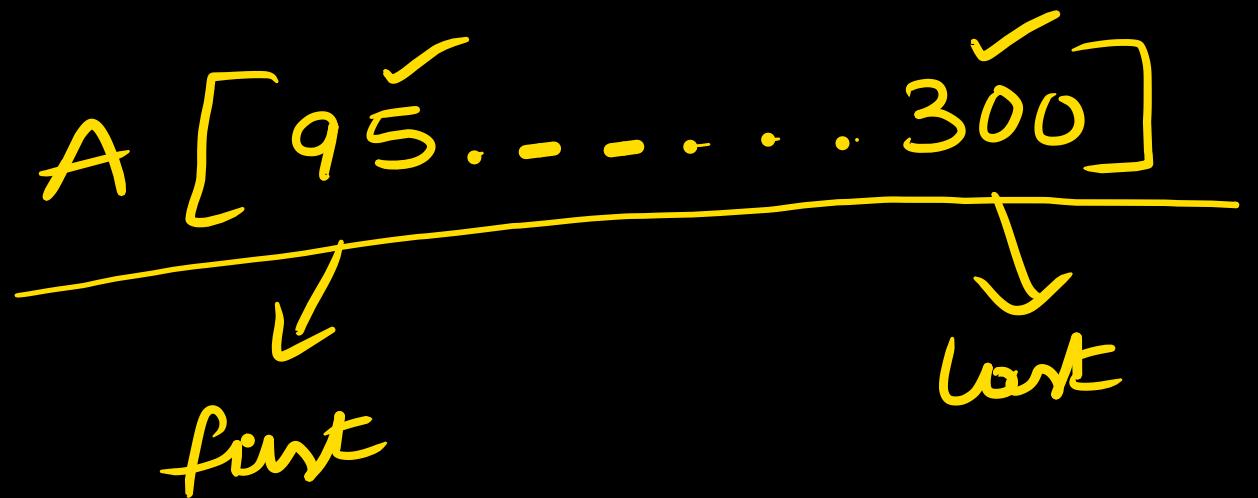
$$\begin{aligned}
 \text{location}(7) &= BA + (7-1) * \underset{\substack{\uparrow \\ \text{size of each element}}}{2} \\
 &= 1000 + 6 * 2 \\
 &= \underline{\underline{1012}}
 \end{aligned}$$

$$o(1 \ 2 \ 3 \ 4 \ \dot{5})$$
$$(5-1) = 4 + 1 = 5$$

$$(5-1) + 1$$

$$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$
$$100 \ \dots \ 200$$
$$(200-100+1)$$
$$7-3+1=5.$$

Gate



$$\text{Total elements} = 300 - 95 + 1 = 206$$

$$\text{size of each element} = 5$$

$$BA = 1000$$

↓  
first element address

loc(A  $\boxed{278}$ )

( $95 \ 96 \ 97 \ \dots \ 277$ )  $\overset{\checkmark}{278}$

$$\underline{277 - 95 + 1} = \underline{\underline{278 - 95}}$$

$$\begin{aligned} & BA + \underline{\underline{(278 - 95) * 5}} \\ & 1000 + \underline{\underline{(278 - 95) * 5}} \\ & = 1915. \end{aligned}$$

$$A [-300 \dots +500]$$

$BA=0$  Size of each element = 10  $\text{loc}[A(-4)]$

$(-300, -299, -298 \dots -35) (-4) -3 -2 -1 0 \dots$

we have to cross all these elements

$$6 + (-4 - (-300)) * 10 = \underline{\underline{2960}}$$

6

$$\begin{aligned} & -5 - (-300) + 1 \\ & = \underline{\underline{-4 - (-300)}} \checkmark \end{aligned}$$

Formula:

$$A[\underline{lb}, \dots, \underline{ub}]$$

BA, size

location  $A[i]$

$$\text{loc}(A[i]) = BA + (i - lb) * \text{size}$$

formula:

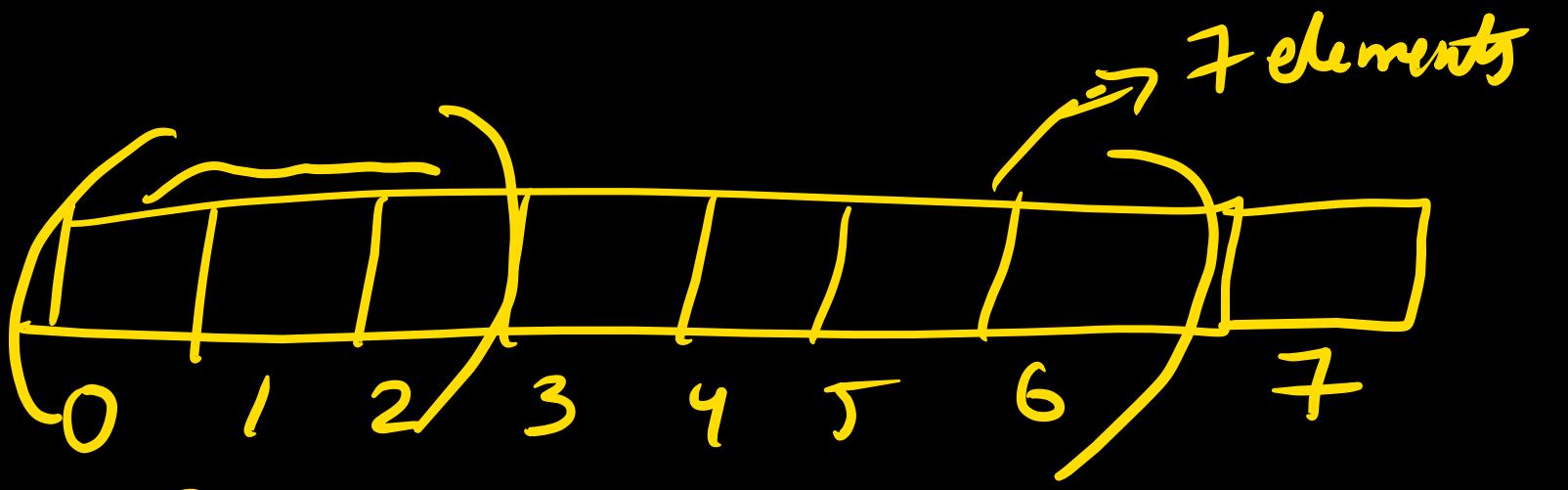
Why every programming language has starting index as 0?

$$BA + \underbrace{(i - lb)}_{\substack{\leftarrow \\ 0}} * \text{size}.$$

lb is always zero if starting add is '0'.

$$BA + i * \text{size}$$

No need of subtraction  
 $\therefore$  less time is taken..



$$BA = 1000$$

$$\text{loc} = \underline{\underline{7}}$$

$$\text{loc}(7) = 1000 + \underbrace{(7-0)}_{\text{size}} * \text{size}.$$

$$\text{loc}(3) = 1000 + \underbrace{(3-0)}_{\text{size}} * \text{size}.$$

t

2D arrays:

C prog is not required for DS.

int a [ ] [ ] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120 }

not filled  
not allowed.

12 × 1  
2 × 6  
3 × 4  
4 × 3  
6 × 2  
1 × 12 } possibilities.

`int a [] [3] = { 10, 20, 30 ..... 120 }`

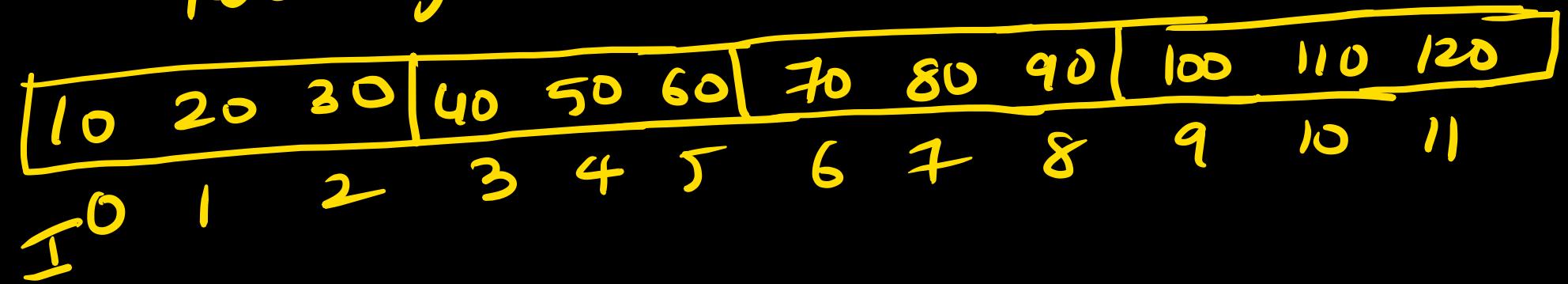
3 columns

4 rows 3 columns

$$\frac{12 \text{ elements}}{3 \text{ col}} = 4 \text{ rows}$$

<i>a</i>	1	2	3
1	10	20	30
2	40	50	60
3	70	80	90
4	100	110	120

Row major order



Cmo



How do you read it.

2D array has to be stored in 1 dimensional array

Rmo Cmo  
loc a[3][3]

Row major order:

	1	2	3	4
1	10	20	30	40
2	50	60	70	80
3	90	100	110	120
4	130	140	150	160

$$\text{BA} = 1000 \\ \text{say } = 2$$

location of  $A[4][3]$

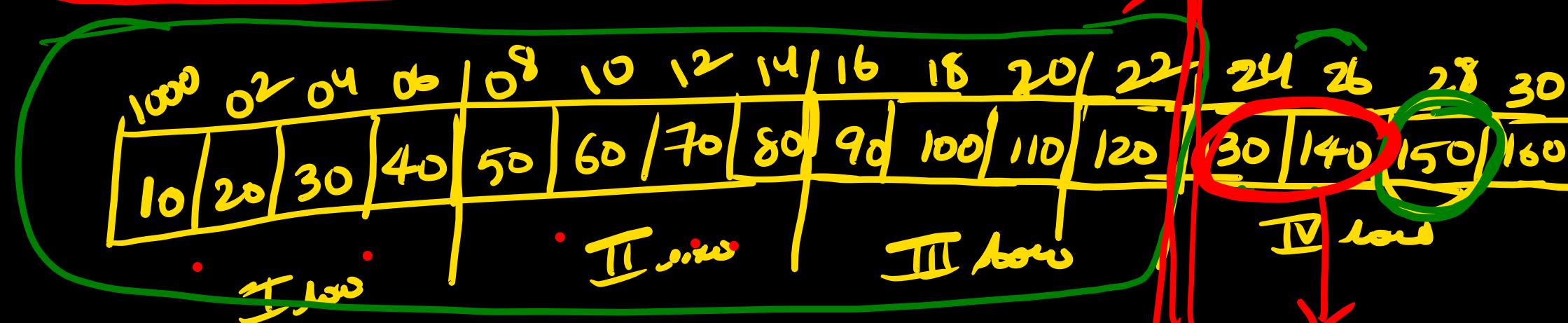
$$\text{BA} \\ 1000 +$$

Ans  
3 rows

$$((4-1) \times 4 + (3-1)) \times 2$$

Ans  
rows

$$(4-1) \times 4$$



Ans  $A[1.....4, 1.....4]$

$$4-1+1=4 \quad 1+1=2$$

$$(3-2+1)$$

$A[27, \dots, 71] [43 \dots 120]$



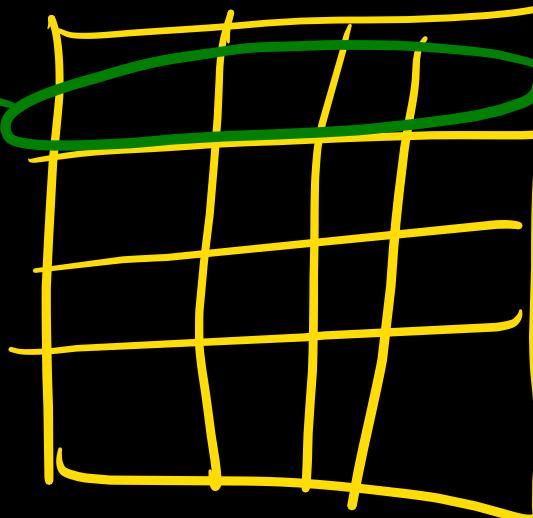
$\downarrow nr$

$$(71 - 27 + 1) \\ 45$$

$\downarrow nc$

$$(120 - 43 + 1) \\ 78$$

No of  
columns



Size of each row = 78 elements

$$BA = 1000$$

size = 10, RMO, bc ( $A(50) [110]$ )

27  
28  
29  
.  
49  
50

$$BA + [(50-27) * 78 + (110-93)] * 10$$

↓

$$1000 = \underbrace{19610}$$


$$\text{ej: } A[-93, \dots +100] [-25 \dots 25]$$

$$BA=0 \quad \text{Sign}=1 \quad \text{RMD} \quad A[95][20]$$

$$0 + ((95 - (-93)) * 51 + 20 - (25)) * 1$$

$$= 9633$$

Formula for RMO:

$A [lb_1, \dots, ub_1, lb_2, \dots, ub_2]$

BA , size , RMO

loc  $A[i][j]$

$$= BA + ((i - lb_1) * nc + (j - lb_2)) * size$$

number of  
columns

size of each row

Album mejor año:

	1	2	3	4	5	
1	11					
2	21	12	22	13	14	
3	31	32	33	34	35	25
4	41	42	43	44	45	

$A[3][5]$

2

cmo

00	02	04	06	08	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38
11	21	31	41	12	22	32	42	13	23	33	43	14	24	34	44	15	25	35	45

I al      II al      III al      IV al      V al

$$A \left[ +25, \dots, +97 \right] \left[ -16, \dots, +200 \right]$$

$\swarrow$  nr  
 $97 - 25 + 1 = 73$        $\Downarrow$   
 $200 + 16 + 1 = 217$

$BA = 0$        ~~$S = 5$~~   $S \cdot x = 5$       cmo

loc  $(A[90][180])$

$$-16, -15, -14, \dots, +0, +1, \dots) \stackrel{180}{\equiv}$$

$$BA + \left[ (180 - (-16)) * 73 + (90 - 25) \right] * 5.$$

$$= 21 = \underline{\underline{71865}}$$

$$A[-42, \dots, +42] \stackrel{\approx}{=} [-84, \dots, 84]$$

$\parallel$   
 $n_r = 85$

$\downarrow$   
 $n_c = 169$

$$BA = 1000, \text{ size} = 10 \quad \text{cmo}$$

location  $[A[-3][-4]]$

$$1000 ( (-4 - (-84)) * 85 + -3 - (-42) ) * 10$$

$$= 69390$$

=

$$A((b_1 \dots \cup b_1) (\textcircled{1} b_2) \dots \cup b_2)$$

BA, size, cmo

loc  $A[i][j]$



$$\text{BA} + [(j - lb_2) * nr + (j - lb_1)] * \text{size}$$

~~$A[i][j]$~~



$\text{ub}_1 - lb_1 + 1$

In gate exam And interview Exams only 2D are asked.

Should we discuss 3D at first.  
=

5 min break;

why to risk.  
==

# 3D arrays:

int a[3....9] [5....15] [8....24]  
↓  
7  
20 array  
↓  
11 rows  
↓  
17 Count  
loc[a[7][12] [20]]

RmD

$$R \text{ m}^2$$

$$1000 + ((7-3) * 11 * 17 + ((12-5) * 17 + (20-8)) * 2) = 2758$$

$a[3 \dots 9] [5 \dots 15] [8 \dots 24]$

$BA = 1000$ , size = "2", CMO.

location  $a[7][12][20]$

$1000 + [(7-3)*11*17 + (20-8)*11 + (12-5)]*2$

$= 2774 \checkmark$

$a[-10 \dots +10, -25 \dots 25, -9 \dots +29]$   
 BA = 0      size = 10, RMO       $\downarrow$  location  $(a[8][20][15])$

$$\begin{aligned}
 & 0 + ((8 - (-10)) * 51 * 39 + \\
 & ((20 - (-25)) * 39 \\
 & ((15 - (-9))) * 10) \\
 & \qquad \qquad \qquad \left| \begin{array}{l} \text{cmo} \\ \hline 0 + (8 - (-10)) * 51 * 39 \\ + (15 - (-9)) * 51 \\ + (20 - (-25)) \end{array} \right) * 10. \\
 & \qquad \qquad \qquad = 375810
 \end{aligned}$$

Formula :-  
 $a[lb_1 \dots ub_1, lb_2 \dots ub_2, \dots lb_3 \dots ub_3]$

BA , size , RMO

$loc[A[i][j][k]]$

$BA + [(i - lb_1) * nr * nc + (j - lb_2) * nc$   
 $+ (k - lb_3)] * size.$

Frinde

$$a[lb_1, \dots, ub_1, lb_2, \dots, ub_2, lb_3, \dots, ub_3]$$

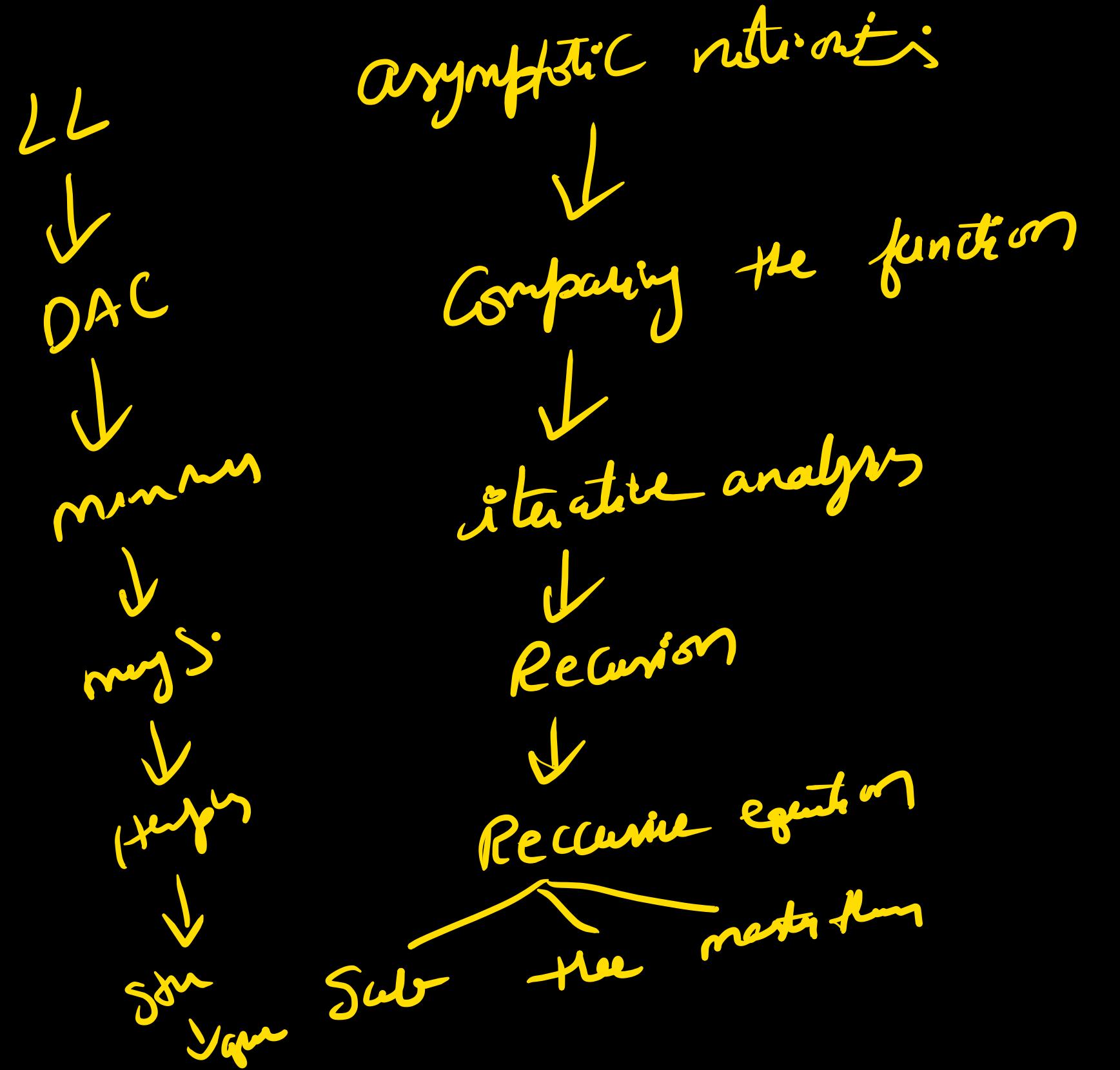
BA, size; ~~CMD~~

$loc(A[i][j][k])$

CMD:

$$BA + ((i-lb_1)*nr*nc + (k-lb_3)*nr + (j-lb_2))*size$$

$(ub_2 - lb_2 + 1) \quad (ub_3 - lb_3 + 1)$



array

$2D <$	$C^{MO}$
$3D <$	$K^{MO}$

Boring part of algos  
over  
linked list  $\rightarrow$  more exciting  
and interesting.