C-Programming and DS

Today Class Topics

Memory Representation and Address calculation of

- 1D Array
- 2D Array
- 3D Array
- Sparse Metrix
  - LTM
  - UTM
- Recursion
- Types of Recursion

Array :-

1) Similar items stored
2) Contiguous memory allocation
3) Random access O(1) time Complexity
4) Array index always starts from 0

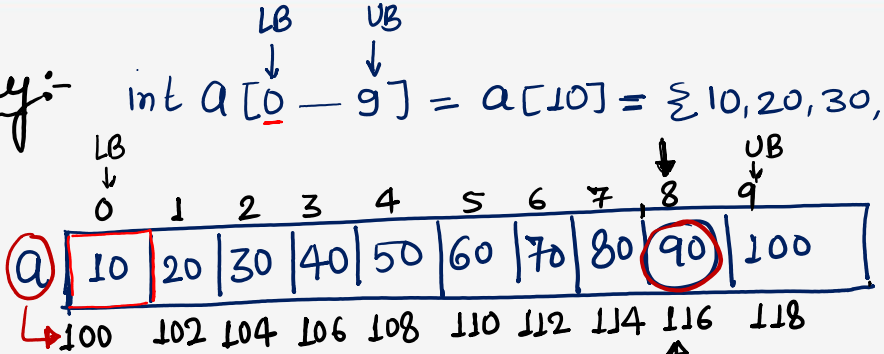
GATE CSE

DS → C

1D-Array:-

int a[0 — 9] = a[10] = {10, 20, 30, ... 100};

int a[10];



Length of Array = UB − LB + 1

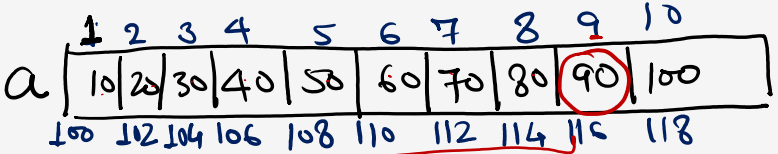
9 − 0 + 1 = 10

SOE = S = Size of each element

int, S ⇒ 2 byte

loc(a[8]) = Base(a) + 8 * SOE ⇒ O(1)

= 100 + 8 × 2 = 116



loc(a[9]) = Base(a) + (9 − 1) * SOE

= 100 + 8 * 2 ⇒ 116 ⇒ O(1)

Question: a[-67 ..... 769], BA = 2000, size of each element = 8 byte, loc(a[636]) = ?

LB = -67, UB = 769

Total elements in the Array = 769 - -67 + 1 = 837

$loc(a[636]) = BA + (636 - -67) * 8 = 7624$

$= 2000 + 7624$

Question: a[-39 ..... +599], BA = 6500, Size of element = 7 byte, loc(a[546]) = ?

$$loc(a[546]) = 6500 + (546 - -39) * 7 = 10595$$

a[lb1 --- ub1], BA, S

$$loc(a[i]) = BA + (i - lb_1) * S$$

$$= BA + i * S$$

NOTE:- a[LB..... UB], BA, Sizeofelement(SOE) = S ,

$$Loc(a[i]) = BA + (i - LB) * S$$

Why array index always start with 0 ?

2D Array :-

Ex: int a[1....4][1....5]

Row: lb_1 to ub_1 ; Col: lb_2 to ub_2

int a[4][5];

Row Col

a[0....3][0....4]

1) No. of Rows (n_r) = $ub_1 - lb_1 + 1 = 4 - 1 + 1 = 4$ ✓

2) No. of Colms (n_c) = $ub_2 - lb_2 + 1 = 5 - 1 + 1 = 5$ ✓

3) 2D-Array will be stored in memory in the forms of multiple 1D Arrays

4) 2D-Array elements will be stored in the memory in 2-different ways

a) Row major order (RMO)

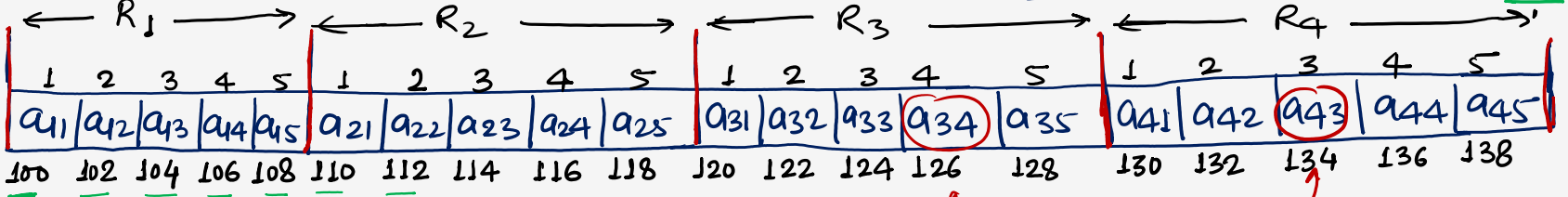
b) Colom major order (CMO)

C-Language by default follows row major order

Row major order (RMO):-

	1	2	3	4	5
1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
⇒ 4	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

4×5

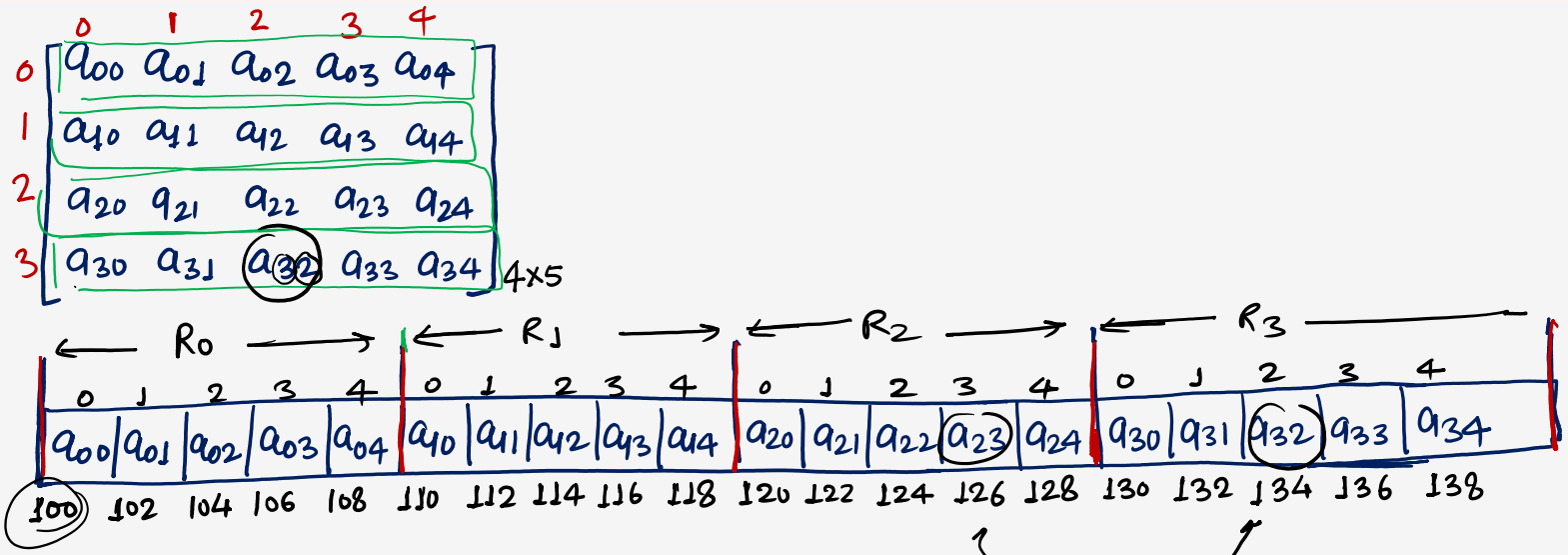


i) $loc(a[4][3]) = ?$ $\quad 100 + [(4-1)*5 + (3-1)]*2 = 134$

(lb_1, lb_2)

ii) $loc(a[3][4]) = ?$ $\quad 100 + [(3-1)*5 + (4-1)]*2 = 126$

(lb_1, lb_2)



i) loc(a[3][2]) = ? $100 + (3*5 + 2)*2 = 134$

$\Rightarrow$ OU)

ii) loc(a[2][3]) = ? $100 + (2*5 + 3)*2 = 126$

Question:- $a[-69 \ldots +398][-19 \ldots +699]$, BA = 7800, S = 9 byte, loc(a[361][673]) = ?

Rows

Soln:-

lb_1 ub_1 lb_2 ub_2

$n_r = ub_1 - lb_1 + 1 = 398 - -69 + 1 =$

$n_c = ub_2 - lb_2 + 1 = 699 - -19 + 1 = 719$

SoE

$$loc(a[361][673]) = 7800 + [(361 - -69) * 719 + (673 - -19)] * 9$$

(lb_1 under -69; lb_2 under -19)

$$= 2796558 \text{ Ans}$$

Question: a[-378 .... +716][-428 .... +658], BA = 8200, SOE = 8

loc(a[699][599]) = ?

n_c = 658 − −428 + 1 =

loc(a[699][599]) = 8200 + [(699 − −(378) * n_c + (599 − −428)] * 8

(699 − −(378) : ↑ lb_1

(599 − −428) : ↑ lb_2

NOTE:- A[lb_1 ..... ub_1][lb_2 ..... ub_2], BA, SOE = S,

(Row, Col)

$n_r = ub_1 - lb_1 + 1$

$n_c = ub_2 - lb_2 + 1$

$$loc(a[i][J]) = BA + [(i - lb_1) * n_c + (J - lb_2)] * S$$

Ex: A[1 .... 10][1 .... 15] of integer. Assume that each integer take one byte, Base(A) = 100,

lb_1 ub_1 lb_2 ub_2 = 15 + 1 = 15

Find loc [A[i][J]] = ?

a) 15i + J + 84

b) 15J + i + 84

c) 10i + J + 89

d) 10J + i + 89

$$= 100 + [(i-1) * 15 + (J-1)] * 1$$

$$= 100 + 15i - 15 + J - 1$$

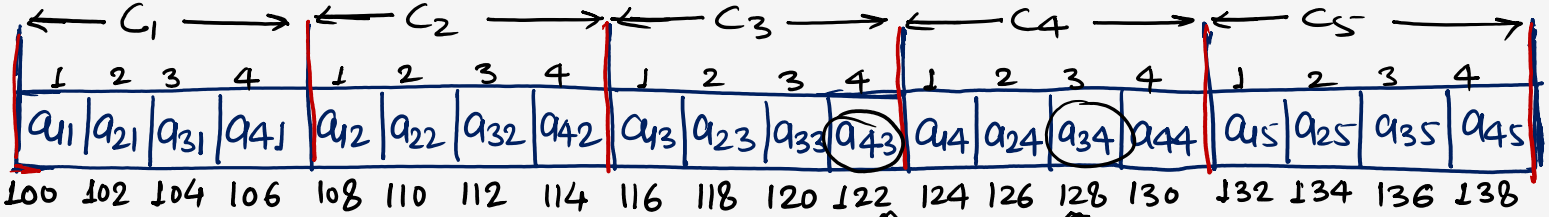
$$= 100 + 15i + J - 16$$

$$15i + J + 84$$

Column major order (CMO):-

	1	2	3	4	5
1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
4	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

4 × 5



i) loc(a[4][3]) = ? $100 + [(3-1) \times 4 + (4-1)] * 2 = 122$

(lb_2, lb_1)

ii) loc(a[3][4]) = ? $100 + [(4-1) * 4 + (3-1)] * 2 = 128$

(lb_2, lb_1)

Question:- a[-29 ... +647][-3 ... +799], BA = 3500, S = 8, loc(a[611][738]) = ?

$$loc(a[611][738]) = 3000 + [(738 - -3) * (647 - -29 + 1) + (611 - -29)] * 8$$

$$= 4021876$$

Question:- a[-311 .... +617][-428 .... +618], BA = 6800, SOE = 7, loc(a[599][577]) = ?

$$loc(a[599][577]) = 6800 + [(577 - -428) * (617 - -311 + 1) + (599 - -311)] * 7$$

$ub_1 - lb_1$; lb_1 ; lb_2

$$= 6548685$$

NOTE: $A[lb_1 \cdots ub_1][lb_2 \cdots ub_2]$, BA, SOE = S,

Row Col

$$n_r = ub_1 - lb_1 + 1$$

$$n_c = ub_2 - lb_2 + 1$$

$$loc(A[i][j]) = B.A + [(j - lb_2) * n_r + (i - lb_1)] * S$$

Ex:- A[1...10][1...15] of integers. Assume that each integer take one byte, Base(A) = 100, find loc[A[i][j]).

a) 15i+j+84
b) 15j+i+84
c) 10i+j+89
d) 10j+i+89
e) None

$= 100 + [(j-1)*10 + (i-1)]*1$

$= 100 + 10j - 10 + i - 1$

$\Rightarrow i + 10j + 89$

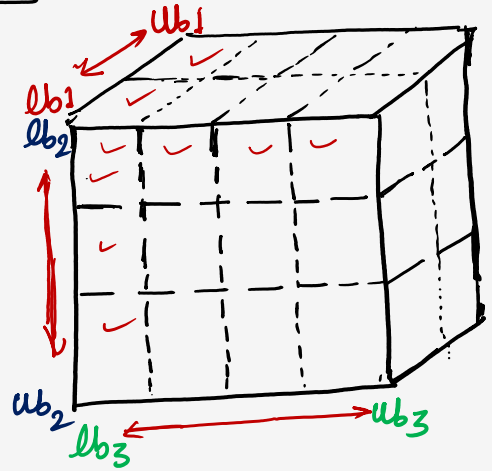
$10j + i + 89$

3D-Array:-

$a[lb_1 \cdots ub_1][lb_2 \cdots ub_2][lb_3 \cdots ub_3]$

Planes or Surfaces or Layers — Row — Col

int [2][3][4]; ⇒ [ ]3×4



$\text{no. of layers}(n_a) = ub_1 - lb_1 + 1$ ✓

$\text{no. of rows}(n_r) = ub_2 - lb_2 + 1$ ✓

$\text{no. of Colms}(n_c) = ub_3 - lb_3 + 1$ ✓

3D-Array stored into multiple 2D-Arrays

Row major order (RMO):-

$8-3+1=6$ $\quad 15-5+1=11$ $\quad 10-4+1=7$

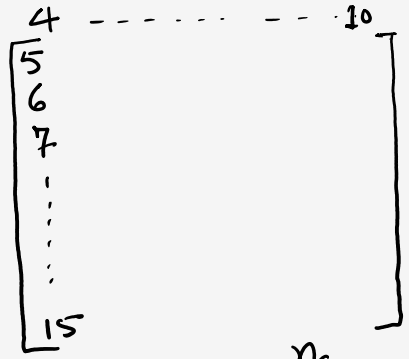
$a[3\cdots 8][5\cdots 15][4\cdots 10]$

6 - 2D Arrays each of size 11×7

11 - 1D Arrays each of size 7

2D

3 - 11×7
4 - 11×7
5 - 11×7
6 - 11×7
7 - 11×7
8 - 11×7



Ex: let BA = 100, S = 2, loc(a[6][8][7]) = ?

$100+[(6-3)*11\times 7+(8-5)\times 7+(7-4)]*2$

Extra Add n_c $\quad lb_3$

lb_1 $\quad n_r$ $\quad lb_2$

$=610$

Ex:

$loc(a[7][14][9]) = 100+[(7-3)*11\times 7+(14-5)*7+(9-4)]*2$

↳ Extra Add

$= 852$

NOTE:- a[lb1 ... ub1][lb2 ... ub2][lb3 ... ub3], BA, SOE = S

Layers Rows Colums

RMO:-

$$loc(a[i][j][k]) = BA + [(i - lb_1) * n_r * n_c + (j - lb_2) * n_c + (k - lb_3)] * S$$

RMO

Extended

CMO

CMO:

$$loc(a[i][j][k]) = BA + [(i - lb_1) * n_r * n_c + (k - lb_3) * n_r + (j - lb_2)] * S$$

$$n_r = ub_2 - lb_2 + 1$$

$$n_c = ub_3 - lb_3 + 1$$