unacademy

Stack & Queue - Part VI

Course on Data Structure

Pankaj Sharma • Lesson 19 • July 29, 2024

unacademy

CS & IT Engineering

Data Structure

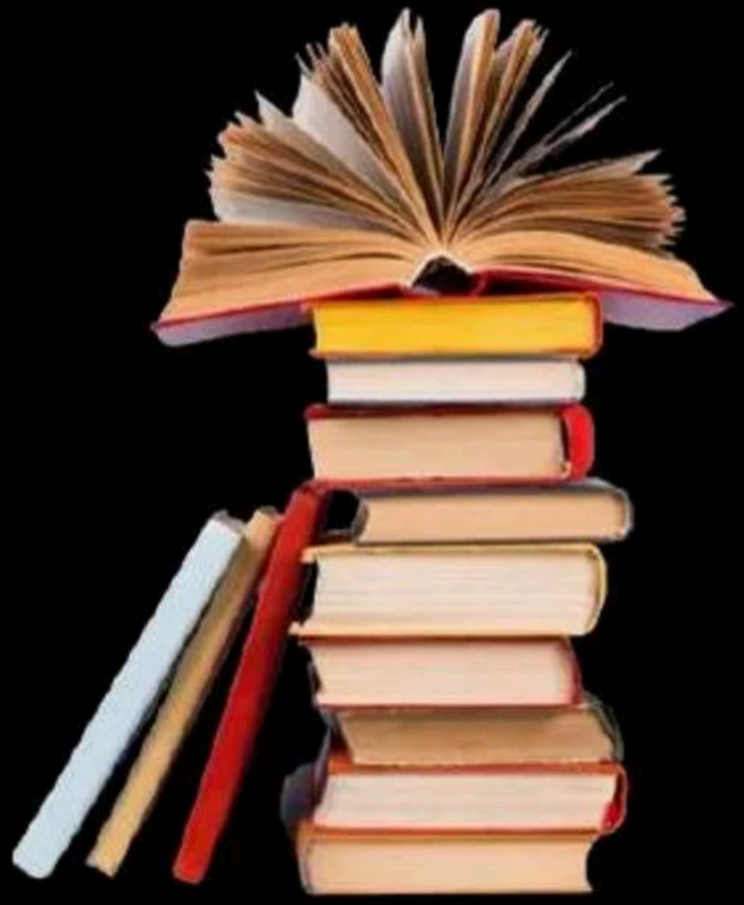
Stack & Queue

Lecture Number- 19

By- Pankaj Sir

Topics

to be covered



1. Satck & Queue

Q1. Consider an integer 2-D array A[-7 to 7][-7 to 7] that stores a UTM U, where

U[i][j] = 1 for all i >= j.

The sum of all the elements in the array is _____

unacademy

$$n = 7 - (-7) + 1 = 15$$

$$\text{non-}\exists < \infty$$

$$15 + 14 + 13 + \cdots + 1$$

$$1 + 2 + 3 + \cdots + 15$$

$$= \frac{15 \times 16}{2} = 120$$

Q2. Consider an integer array that stores a LTM arr[-16..15][-16..15] having base address 1000.

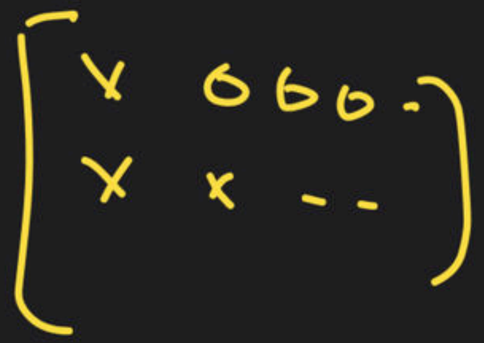
If size of integer = 4 bytes, then the address of element arr[8][7] is ____

RM10

unacademy

Grid [-16..15][-16..15]

RMO



15-(-16)+1 = 32

32x32

q[8][7]

-16 to 7

= 7-(-16)+1

= 24 rows

1+2+3+...24 → 12 / 24x25 = 300

Already within row index 8 are already filled

= -16 to 6

= 6-(-16)+1 = 23

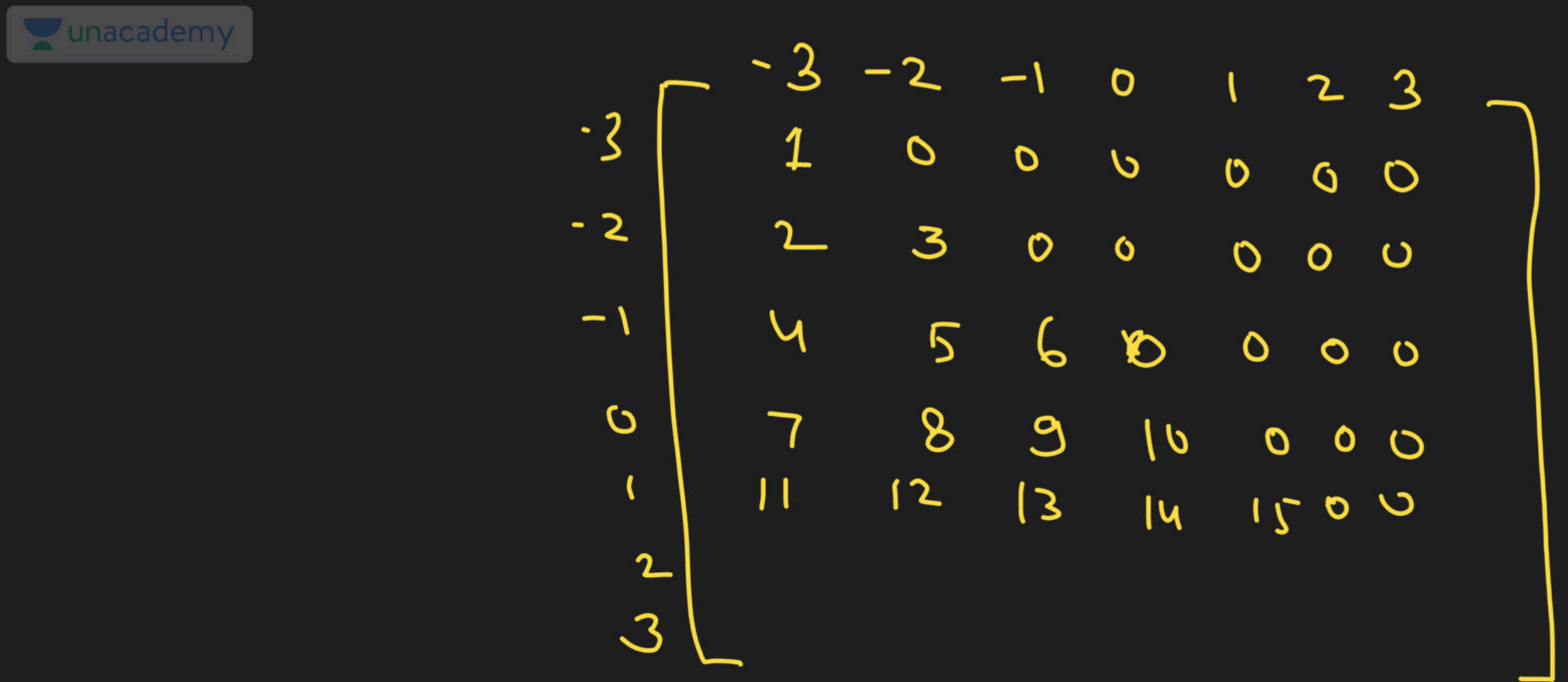
Total elements already filled = 300 + 23

= 323

Total memory already filled = 323 × 4

= 1292 bytes

9] Consider the natural numbers starting from 1 are stored in a LTM arr[-3..3][-3..3].
Find the element present at arr[1][1].


-3 -2 -1 0 1 2 3
-3
-2
-1
0
1
2
3
1 0 0 0 0 0 0
2 3 0 0 0 0 0
4 5 6 0 0 0 0
7 8 9 10 0 0 0
11 12 13 14 15 0 0

-3 to 0

0 - (-3) + 1

= 4

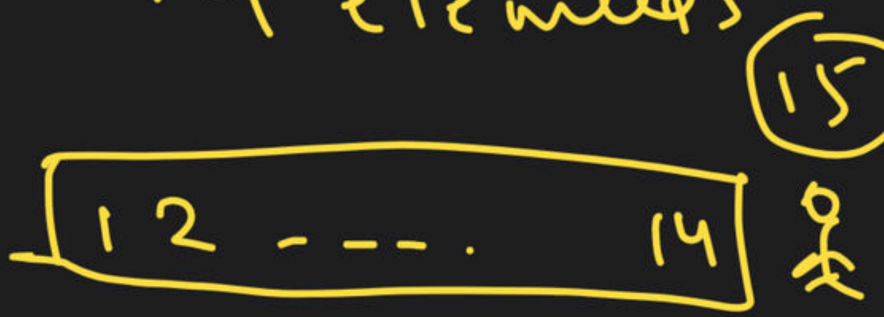
-3 to 6

= 4 elem

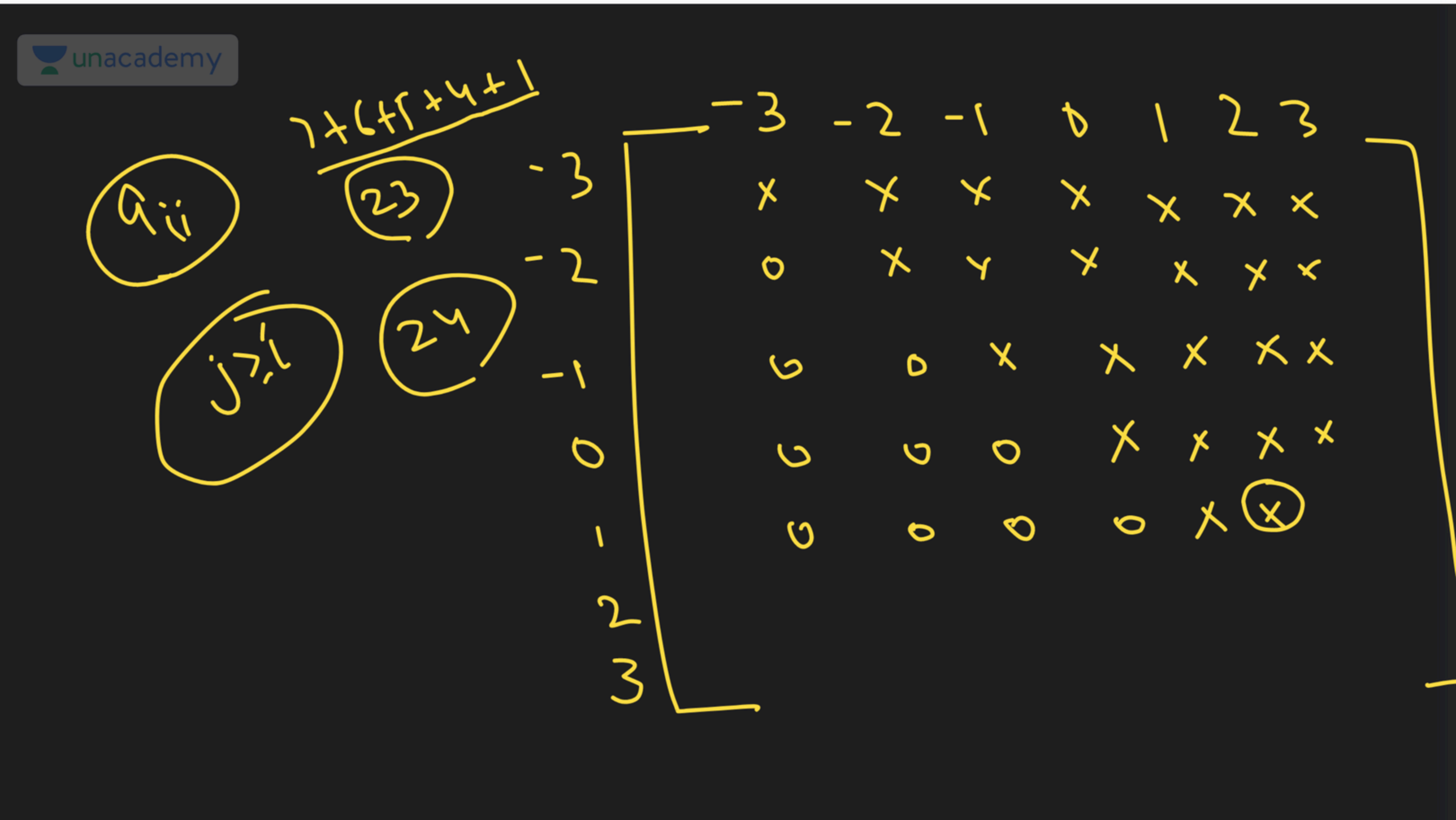
1 + 2 + 3 + 4

= 10

14 elements



9] Consider the natural numbers starting from 1 are stored in a UTM arr[-3..3][-3..3].
Find the element present at arr[1][2].


unacademy
7+6+5+4+1
23
9
ii
24
j ≥ i
-3 -2 -1 0 1 2 3
-3
-2
-1
0
1
2
3

9x9

9) Consider a 2-D array arr[-4..4][-4..4] stores an UTM. Find the address of the ele. a[-1][2], if the starting add. of the array is 500 and size of each ele. is 8 bytes.

a) RMO ✓

b) CMO

unacademy

UTM

9x9

q[-4..4][-4..4]

RMD

q[-1][2]

-4 to -2

-2 - (-4) + 1 = 3 rows

9 + 8 + 7

⇒ 24

Within row index -1

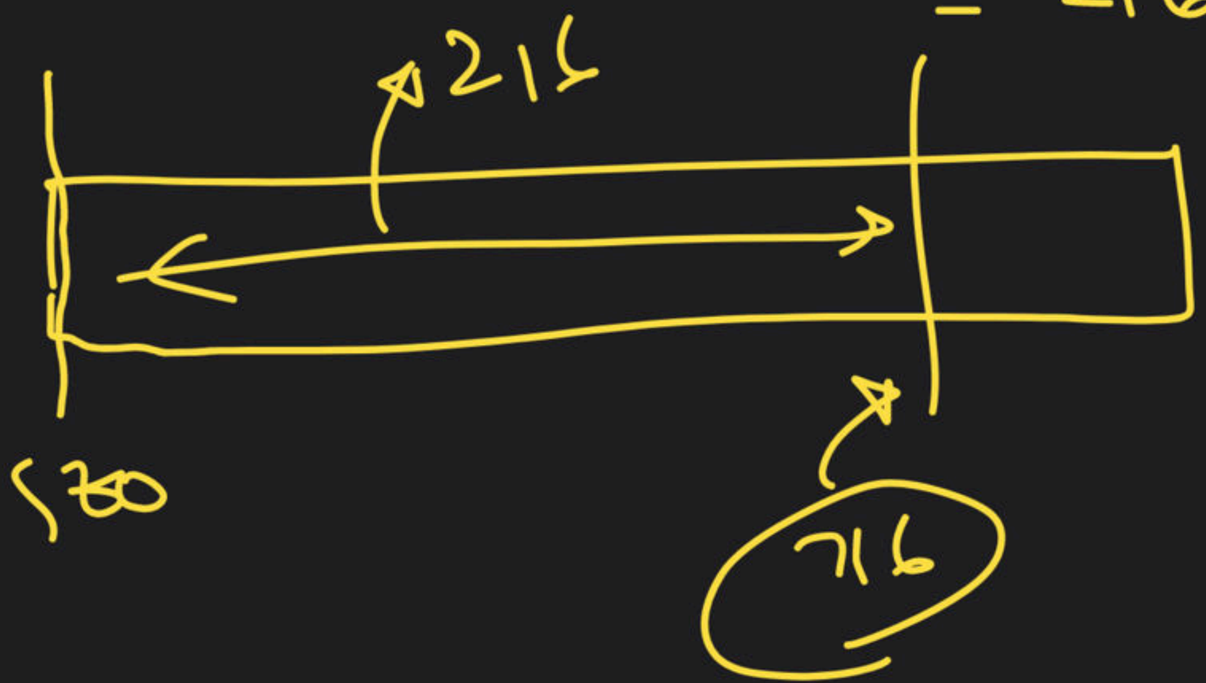
else filled

= 2 - (-1) = 3

Total dc already filled = 27

Memory already filled = 27 × 8

= 216 bytes



CMO

$A[-4..4][-4..4]$

UTM

$Q[-1][2]$

within col index 2

-4 to 1

$1-(-4)+1=6$

$1+2+3+\dots 6$

$= \frac{6\times 7}{2} = 21$

elt =

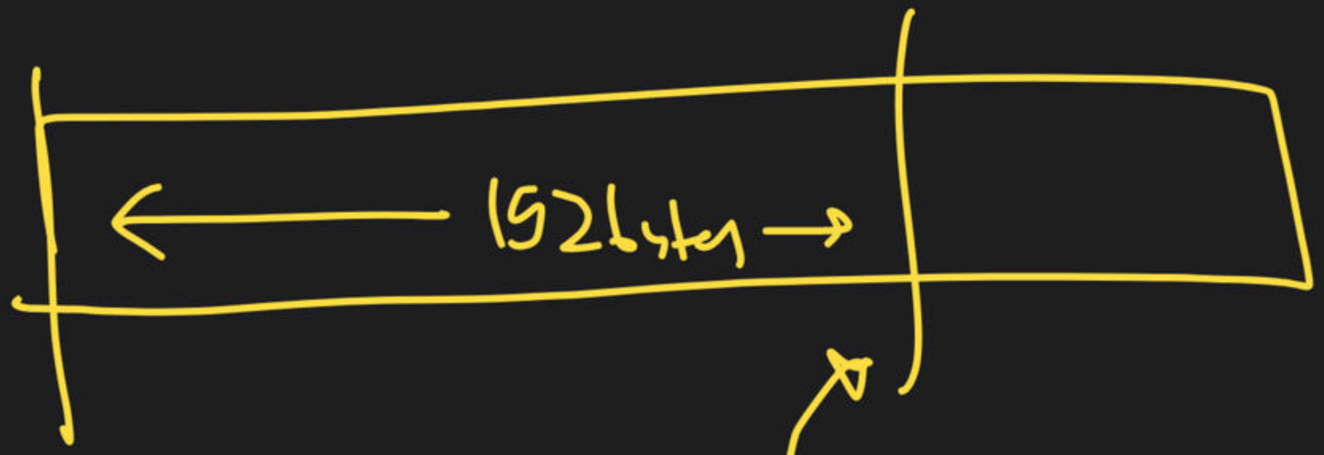
-4 to -2

$= -2-(-4)+1$

$= 3$

Total ele = 24

Memory = 24 × 8 = 192 bytes



Q) Consider a 2D array a[-4..4][-4..4] → ALTM.

a[-2][-1]

BA = 500

w = 8 bytes

CMO

① Anjana - RM

② Suriyesh - BS

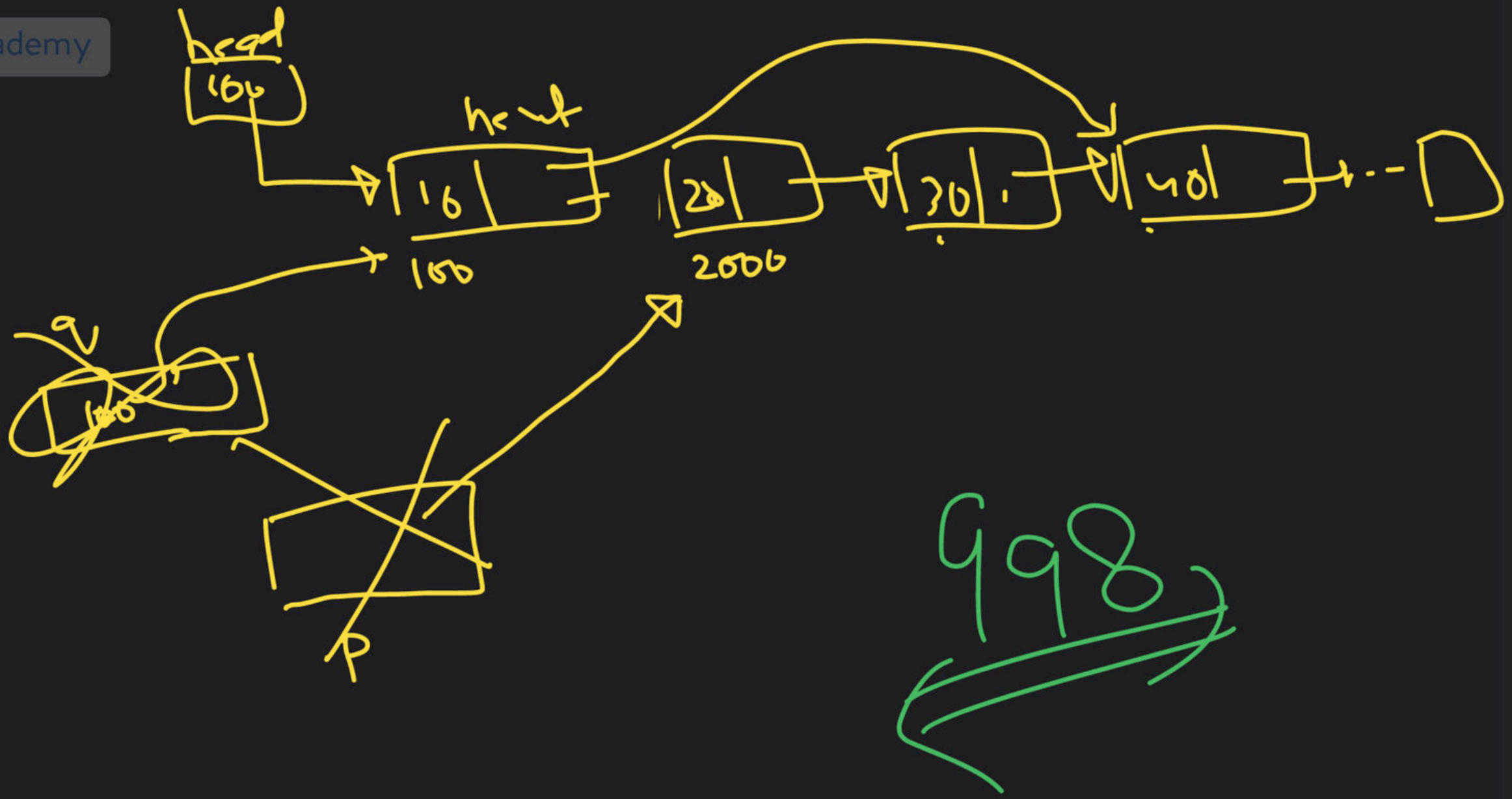
Invalid

Q) Consider a singly linked list q with 1000 elements is passed to the following function:

```
struct node {
  int data;
struct node *next;
};
```

```
void f(struct node *q) {
    struct node *p;
    p = q → next;
    q → next = p → next → next;
```

The size of the LL q after execution of the func.


head
100
next
10
20
30
40
100
2000
q
100
p

Q

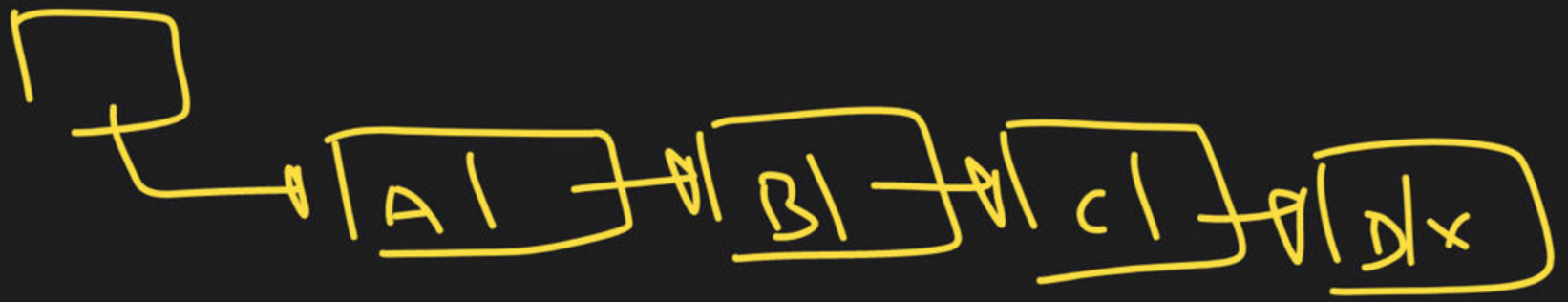

```
void f (struct node * qv) {
    struct node *p;
    p = qv -> next -> next -> next;
    qv -> next -> next -> next = p -> next -> next;
    p -> next -> next = qv -> next;
    pf ("%c", p -> next -> next -> next -> data); }
```

Q) P: LL support Linear accessing of elements.

Q: LL support random accessing.

Incorrect

Q. Consider a singly linked list



is passed to the following function.

```
void f(struct node *qr)
{
  if(qr==NULL) return
     f(qr->next);
     pf("%c", qr->data);
}
```

unacademy

Q1

P: Insertion at the end of L.L. is → O(n) expensive than insertion at the beginning of LL. → O(1)

Q: Deletion at the beginning of LL is cheaper as compared to deletion at the end of LL → O(n)

Both are correct.

1: Given an array of integer, All elements occured even no. of times except one. Find that element.

Traditional → O(nlogn) i/p : {2, 4, 4, 3, 2, 3, 7}

XOR → O(n) o/p : 7

$a \wedge a = 0$

$b \wedge a = a$

$a \wedge a \wedge a = a$

$a \wedge a \wedge a \wedge a = b$

Jai prakash

$O(n^2)$ ✓

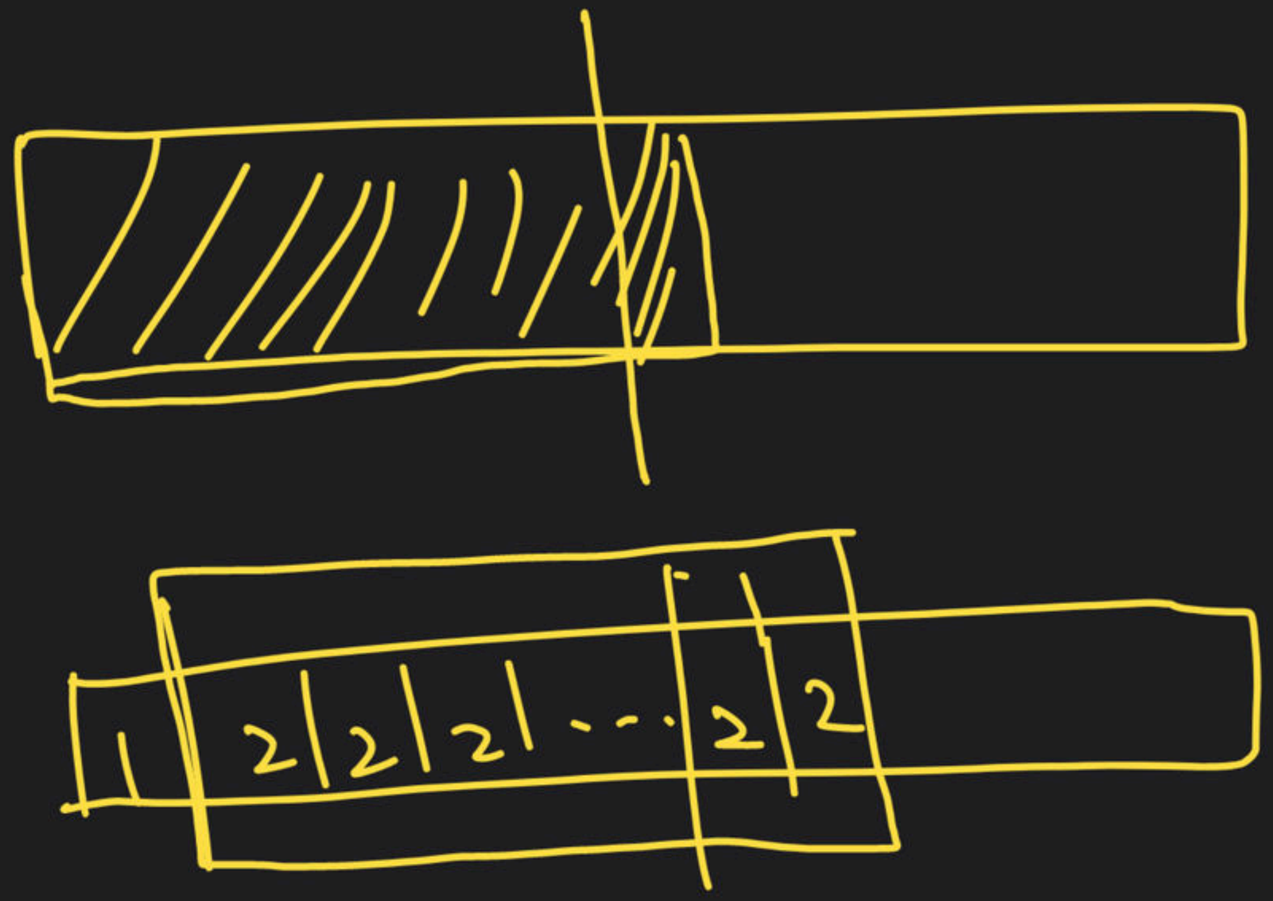
$O(n \log n)$ ✓

$O(n)$ ✓

unacademy

Q2. Given an sorted array that contains 1 majority element (that appears more than N/2 times). Find the majority element.

$O(1)$



Unsorted

1. Moore voting Algo

unacademy

1) Arrays → Revision
2) Linked List
3) Stack

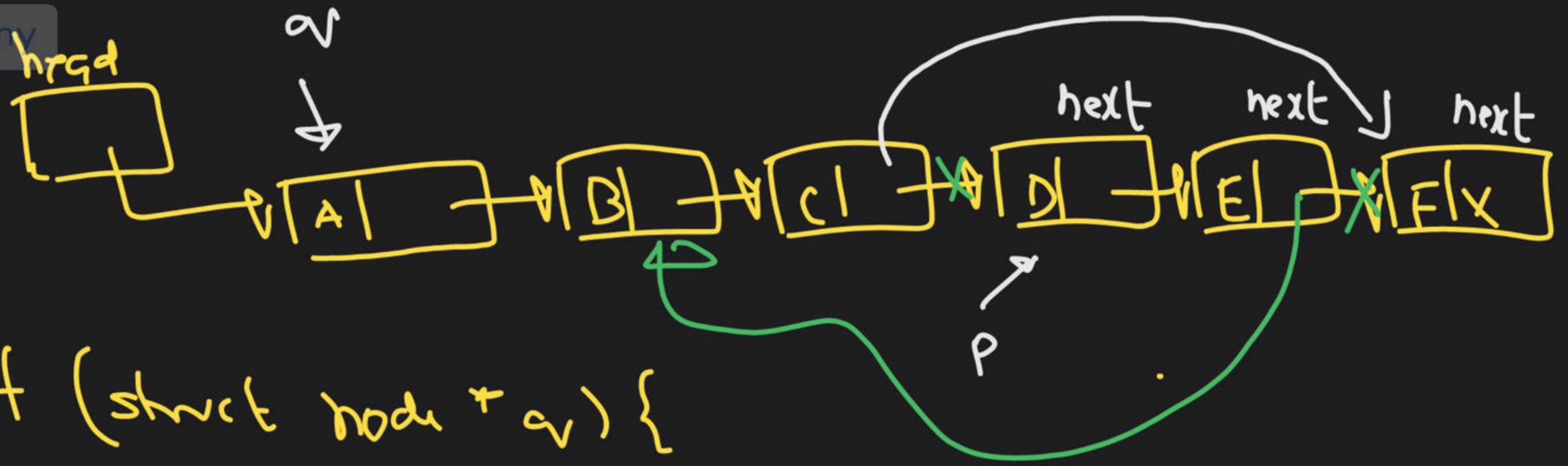
Revision → Special class
→ DS DPDs
→ Must

: [Theory, Concept ⇒ use → attend
apply]

unacademy

০: ০০ - ১১: ০০

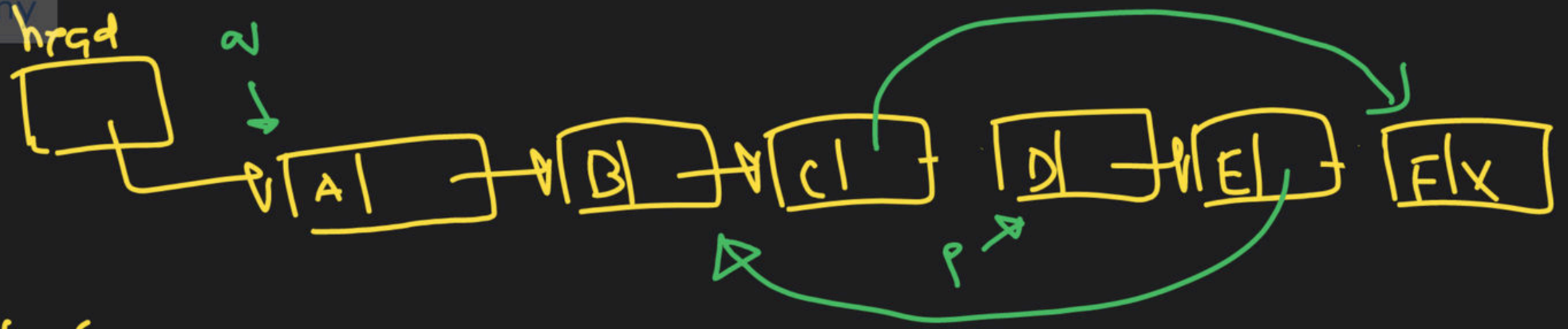
Q




```
void f (struct node * qv) {
    struct node *p;
    p = qv -> next -> next -> next;
    qv -> next -> next -> next = p -> next -> next;
    p -> next -> next = qv -> next;
    pf ("%c", p -> next -> next -> next -> data); }
```

last node

Q



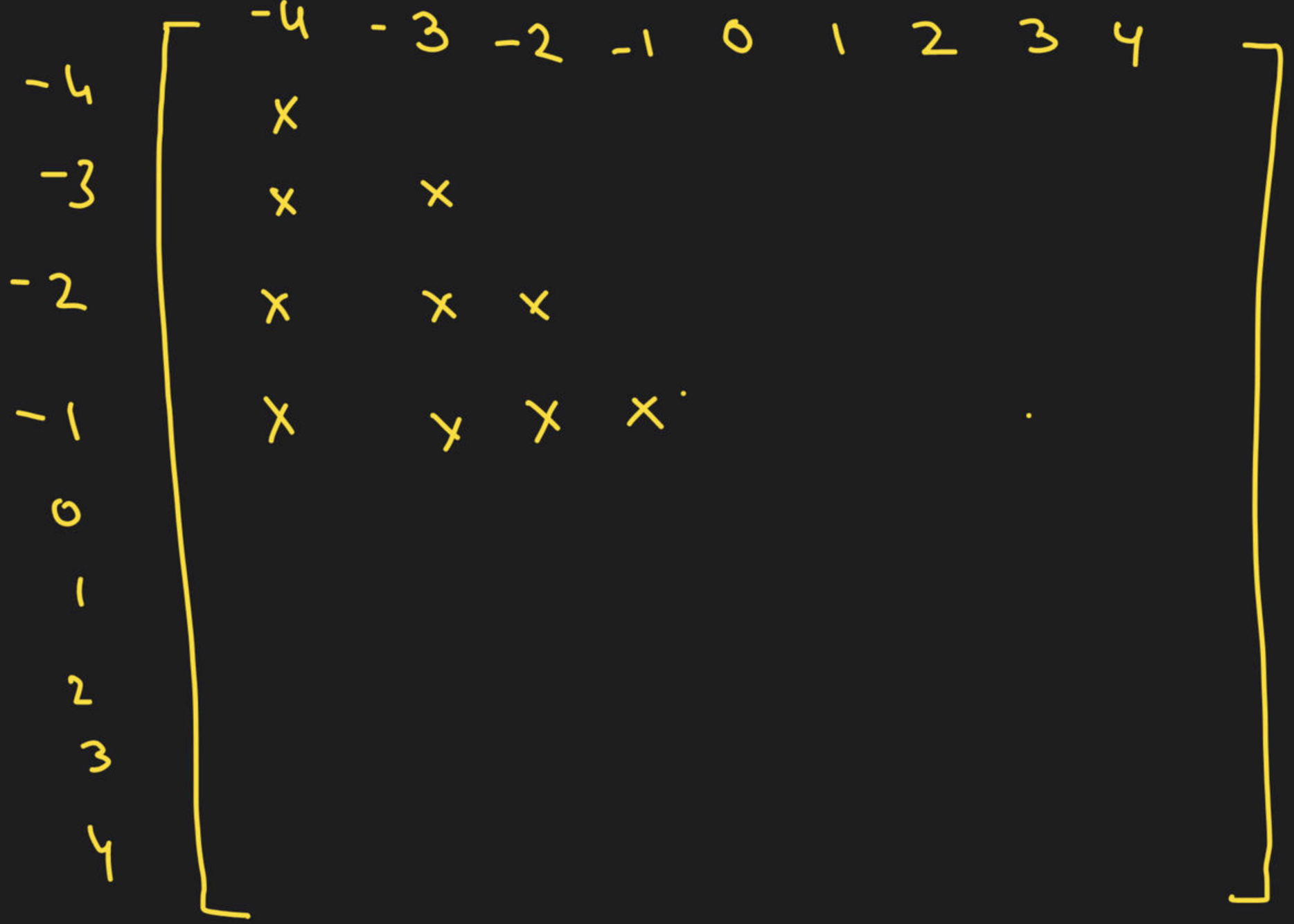

```
void f (struct node * qv) {
    struct node *p;
    p = qv -> next -> next -> next;
    qv -> next -> next -> next = p -> next -> next;
    p -> next -> next = qv -> next;
    pf ("%c", p -> next -> next -> next -> data); }
```

C

Q


```
void f (struct node * q) {
    struct node *p;
    p = q -> next -> next -> next;
  q -> next -> next -> next = p -> next -> next;
    p -> next -> next = q -> next;
 pf ("%c", p -> next -> next -> next -> data); }
```

L 7M



$i == j$

	0	1	2	3
0	a_{00}	0	0	0
1	a_{10}	a_{11}	0	0
2	a_{20}	a_{21}	a_{22}	0
3	a_{30}	a_{31}	a_{32}	a_{33}

unacademy

THANK YOU!

Here's to a cracking journey ahead!