Array Class-2

Special class

Love Babbar • May 19, 2024

→ Target → array → comfortable → Questions

Leetcode → i/p → arr → [a, b, c, c, a, b, d]

find

EK baar

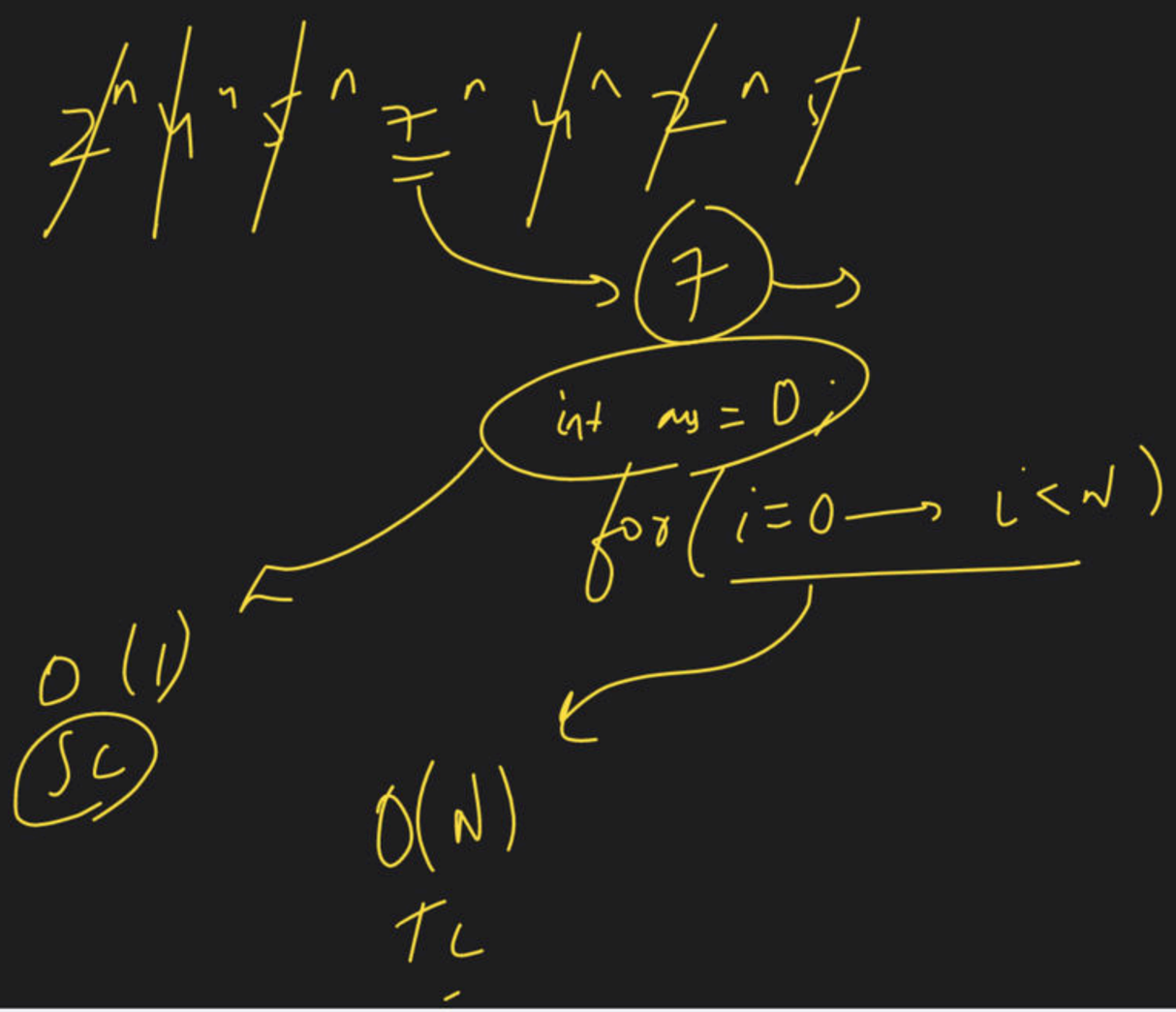
Time complexity	→ Linear	→ O(N)
Space	→ constant	→ O(1)

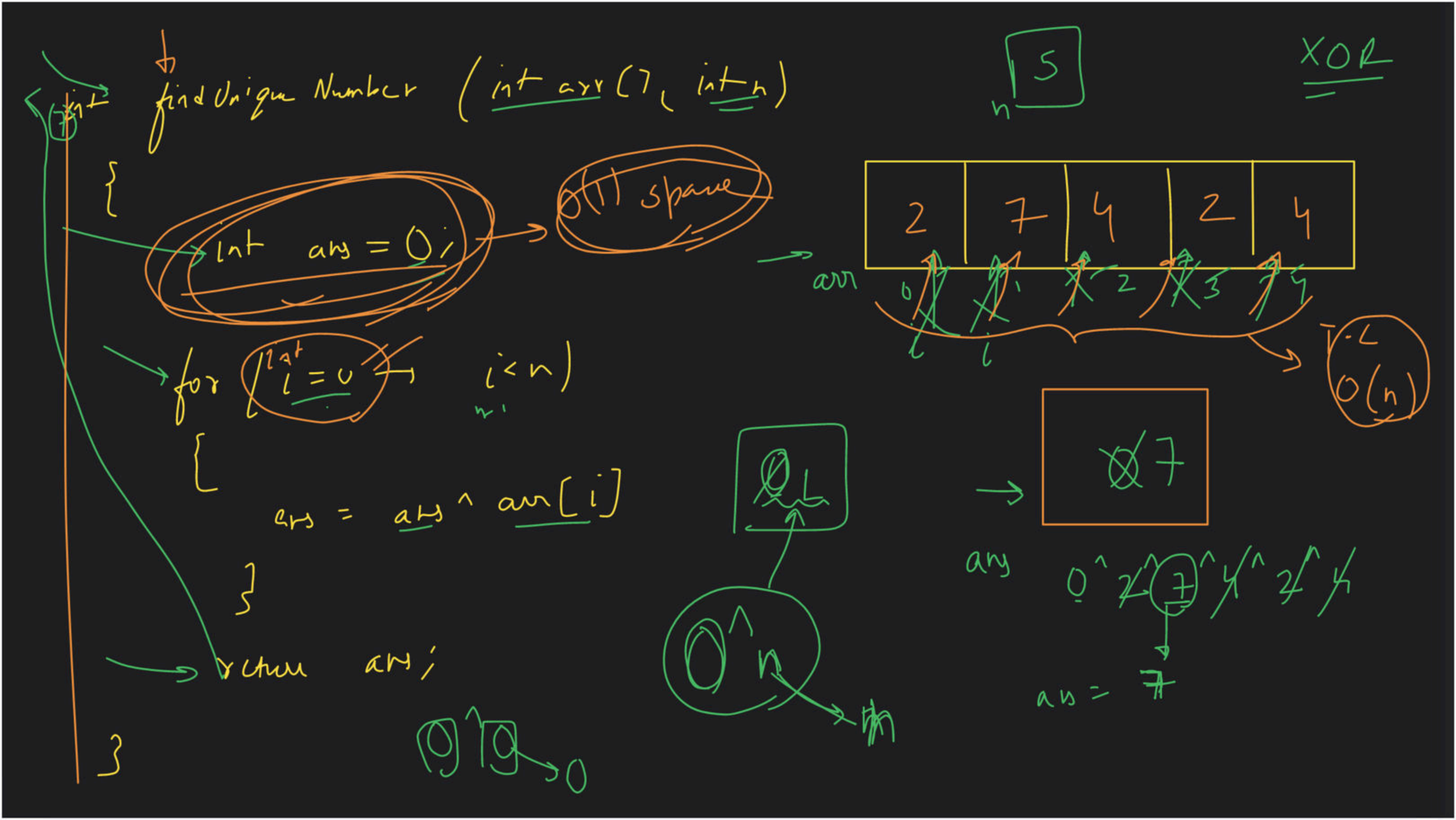
i/p → [2, 4, 5, 7, 4, 2, 5]

xor → a^a

2^2

loop →



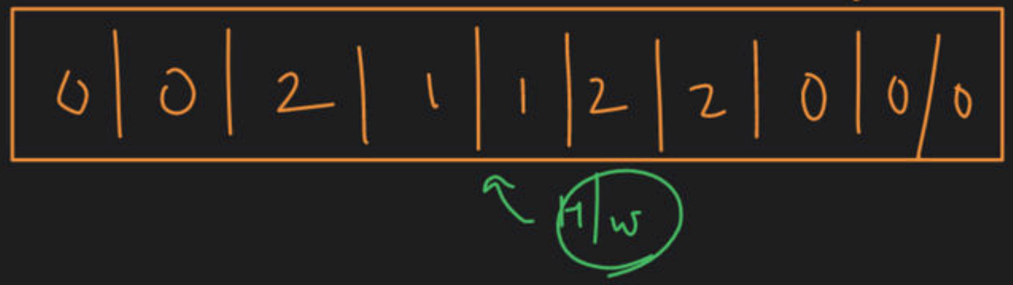

findUniqueNumber (int arr[], int n)
int
{
int ans = 0;
O(1) space
for (int i = 0; i < n)
{
ans = ans ^ arr[i]
}
return ans;
}
n
5
XOR
2
7
4
2
4
arr
0
1
2
3
4
i
i
O(n)
0
0^n
n
ans
0^2^7^4^2^4
7
ans = 7

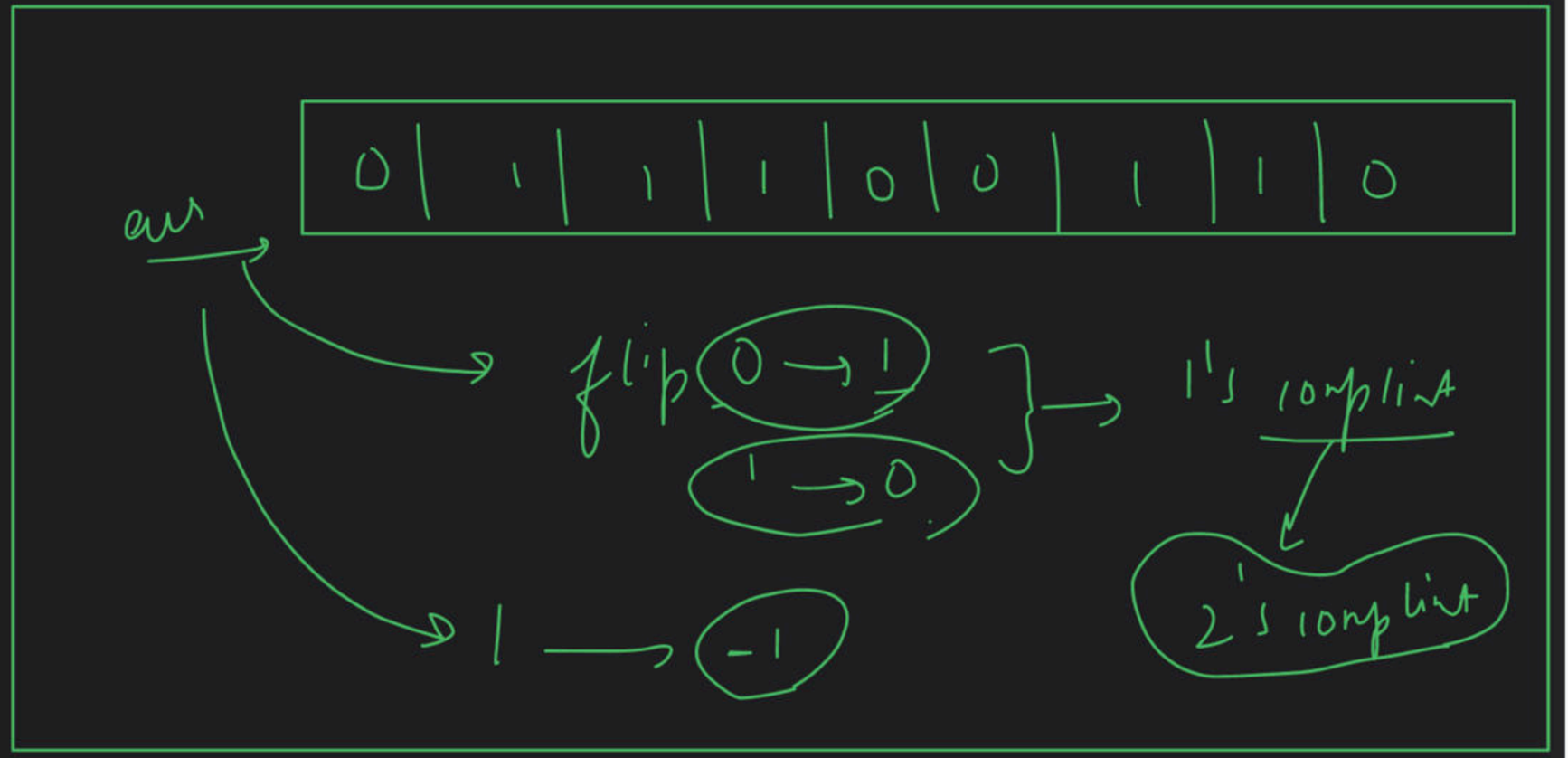
alternate solution → google/gpt / c++

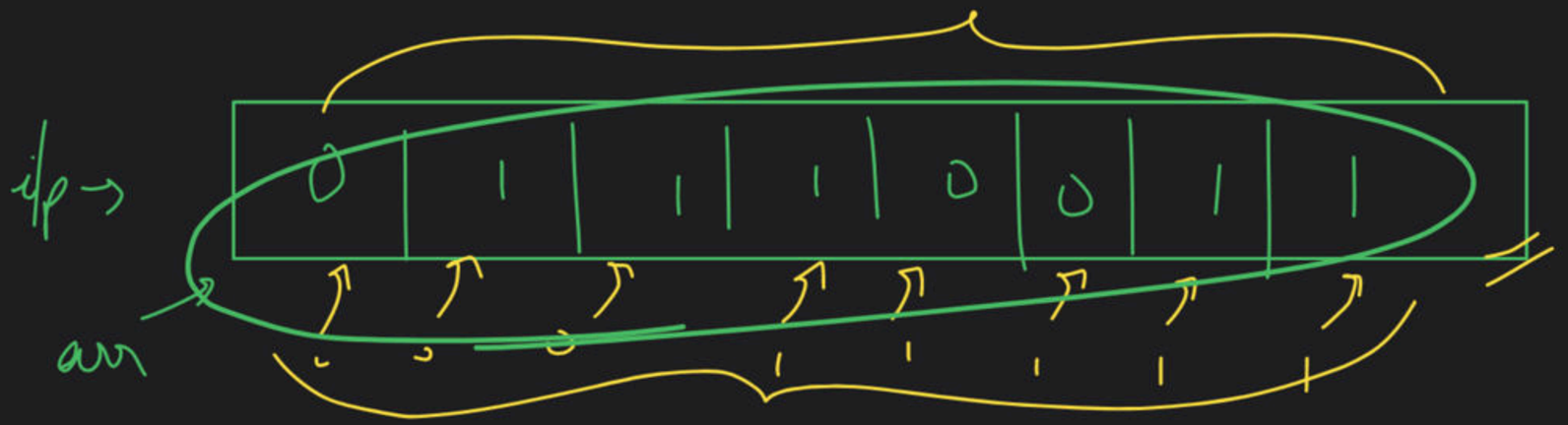
Sort 0's & 1's

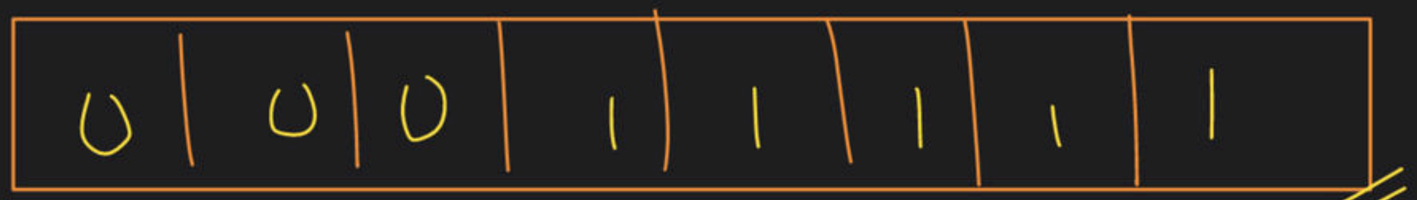
Sort 0's, 1's & 2's

Dutch National flag









algorithm

O(N)
+ O(N)

(1) Counting

zeroCount = 3
oneCount = 5
3 zero inn
5 one inn

2O(N)
O(N)

O(1)
span

30 min

(3) 2 pointer

KHUD

(2) sorting

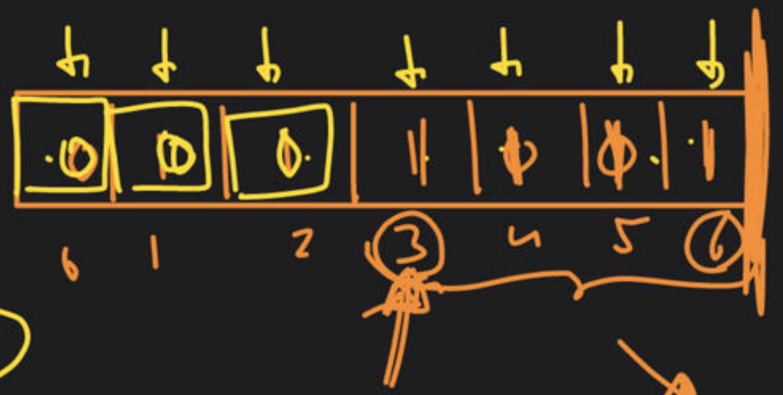
sort (arr, arr+n)

O(nlogn)

asc ady

0 0 0 1 1 1 1 1

```
void SortZeroOne (int arr[], int n)
{
  //Count   int zCount = 0, oCount = 0;
  for (i = 0 → i < n)
  {
    if (arr[i] == 0) → zCount++;
    if (arr[i] == 1) → oCount++;
  }
//insertion
  for (i = 0 → i < zeroCount)        // i < 3
  {
    arr[i] = 0;
  }
  for (i = zeroCount; i < n; i++)
  {
    arr[i] = 1;
  }
}
```



zCount = 3

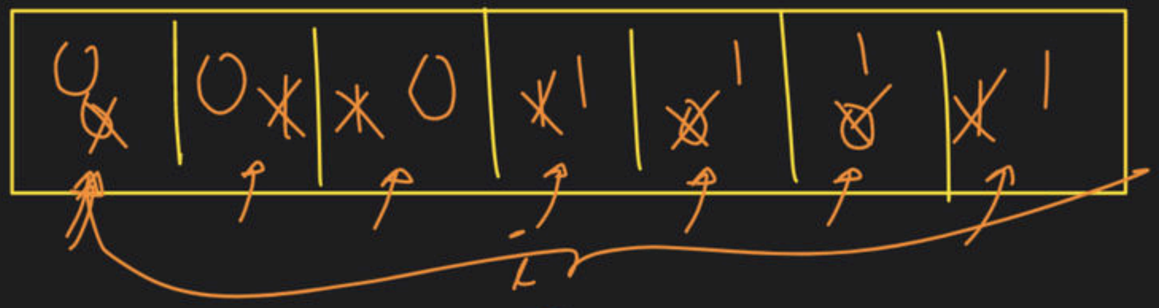
oCount = 4

1 min Break

County

① zeroCount = 0+1+1+1 → 3

oneCount = 0+1+1+1+1 → 4



7 = 3 → 7 → 4

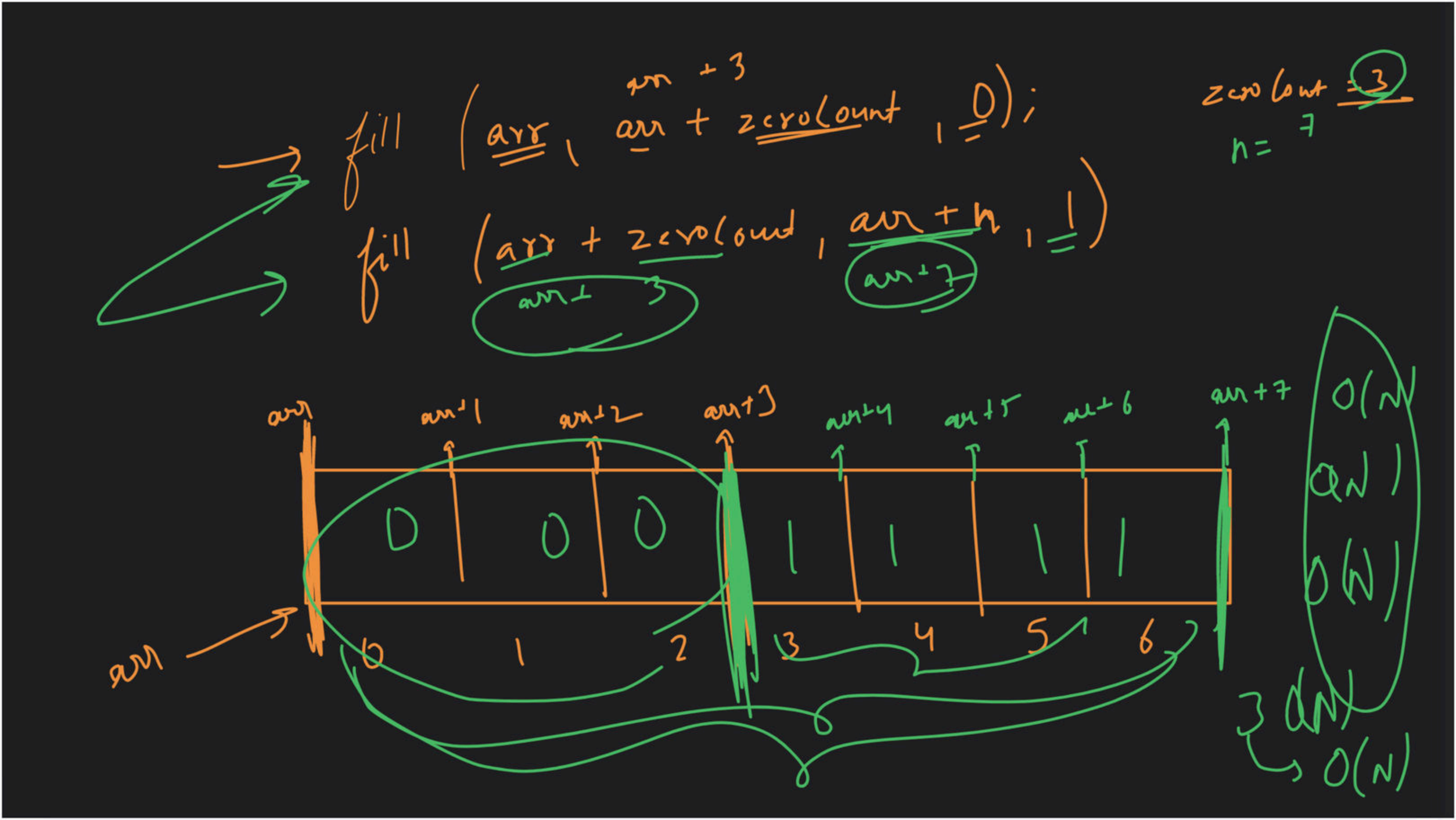
inCounty

zeroCount

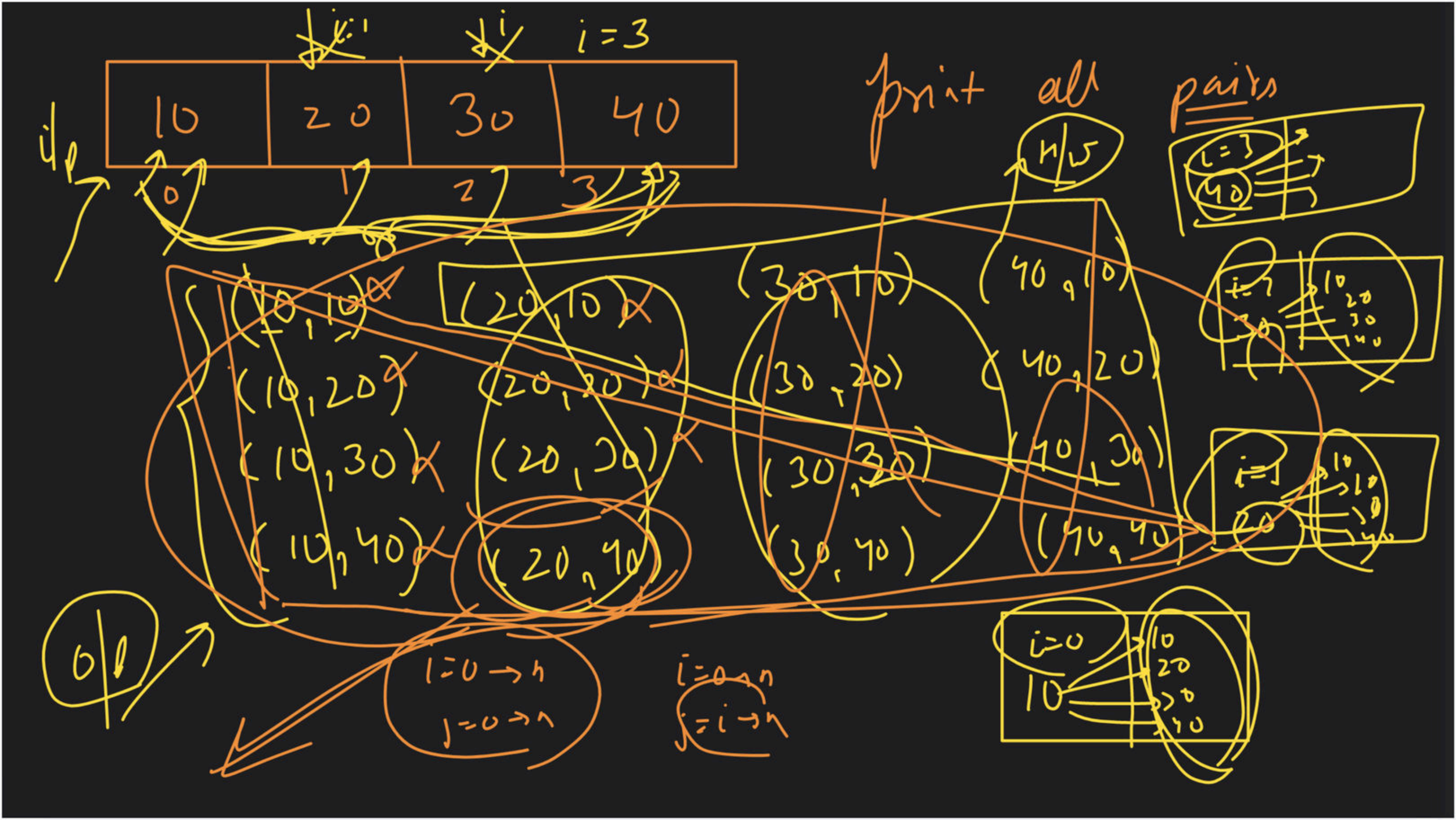
0, 1, 2

```
for(i=0; i<3; i++)
{
    arr[i] = 0
}
```

```
for(i=3; i<n; i++)
{
    arr[i] = 1
}
```

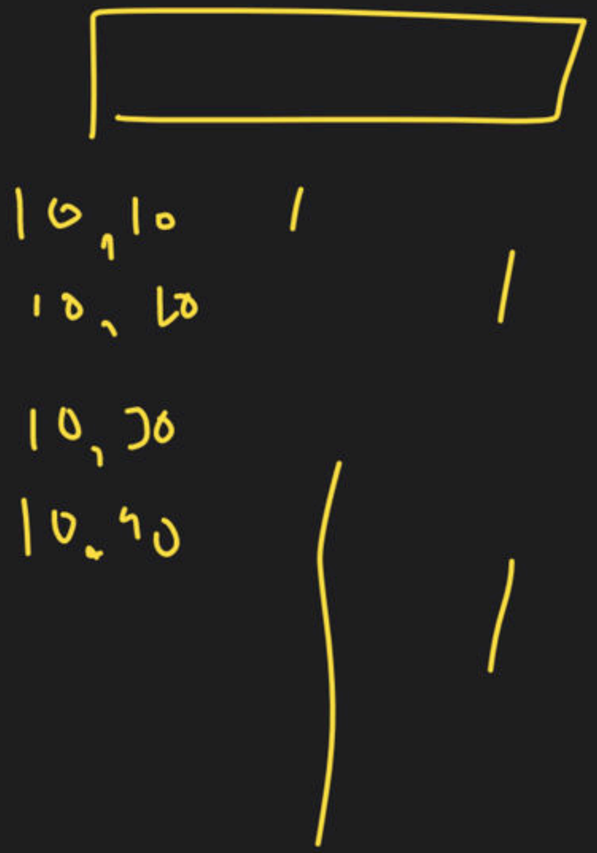

fill (arr, arr + zeroCount, 0);
arr + 3
zeroCount = 3
n = 7
fill (arr + zeroCount, arr + n, 1)
arr + 3
arr + 7
arr
arr+1
arr+2
arr+3
arr+4
arr+5
arr+6
arr+7
0 0 0 1 1 1 1
0 1 2 3 4 5 6
O(N)
O(N)
O(N)
O(N)


L.S
for (i=0 → i<r)
{
}

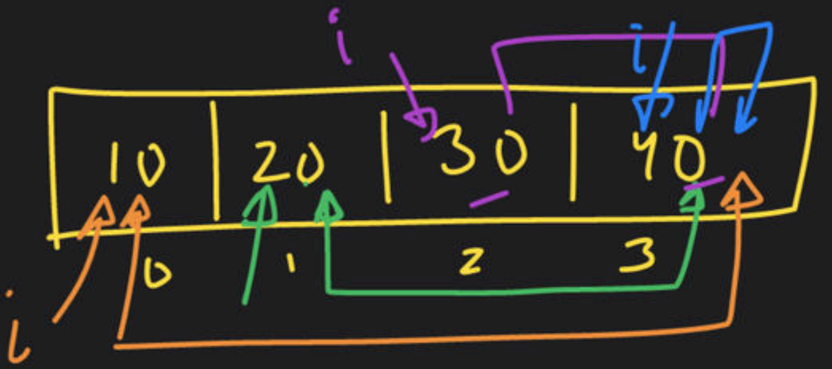

i=1
i
i=3
10
20
30
40
0
1
2
3
i/p
print all pairs
H/W
(10,10)
(10,20)
(10,30)
(10,40)
(20,10)
(20,20)
(20,30)
(20,40)
(30,10)
(30,20)
(30,30)
(30,40)
(40,10)
(40,20)
(40,30)
(40,40)
i=1
i=2
i=3
i=4
o/p
i=0→n
j=0→n
i=0→n
j=i→n
i=0
10
20
30
40

```
for (i → 0 ⟶ <n)
{
    for (j → 0 ⟶ <n)
    {

    }
}
```



```
for ( i = 0 ⟶ n )
{
    for ( j = i ⟶ < n )
    {
    }
}
```



10, 10
10, 20
10, 30
10, 40

20, 20
20, 30
20, 40

30, 30
30, 40

40, 40

all pair

```
for ( i=0 ⟶ i<n )
{
    for ( j=0 ⟶ j<n)
    {
        cout << arr[i] <<arr[j];
    }
}
```

① for (i=0 → n)
{
for (j=i ⟷ n)
{
}
}

② for (i=0 → n)
{
for (j=0 ; j<i ; j++)
{
}
}

③ for (i=0 ⟶ i<n)
{
for (j=0 → j<=i ; j++)
{
}
}

④ for (i=0 → i<n)
{
for (j=n-1 ; j>=0 ; j--)
{
}
}

① for (i = 0 ⟶ n)

{

for (j = n-1 ; j > i ; j- -)

{

}

}

Code → o/p

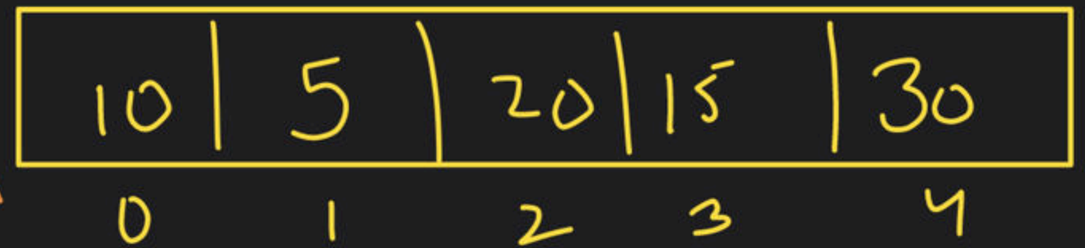
by Understand

0

```
for (i=0 ⟶ i<n)
{
    for (j=i+1 ; j<n; j++)
    {
    }
}
```

Two Sum :-

i/p →



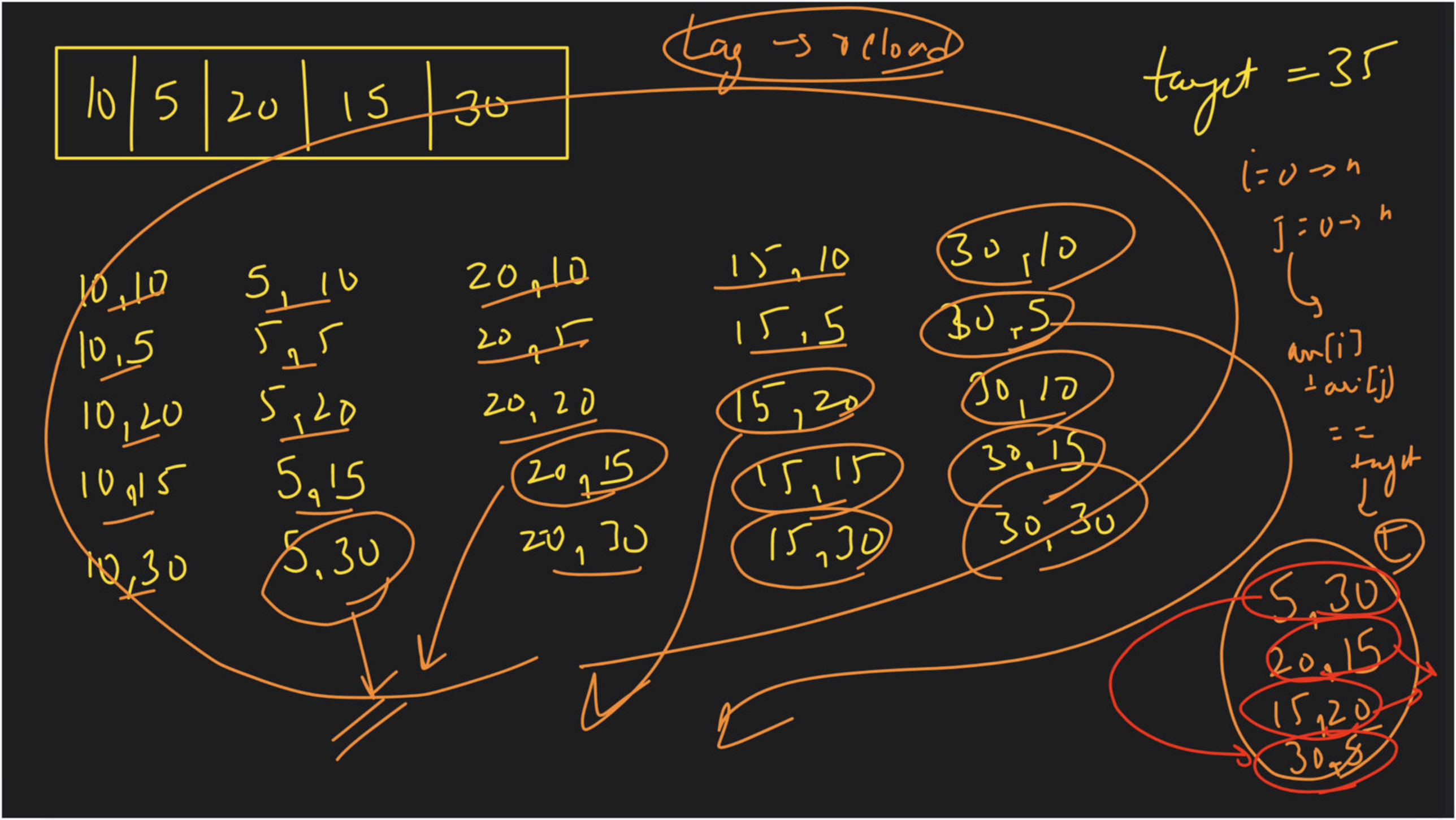
target → 35

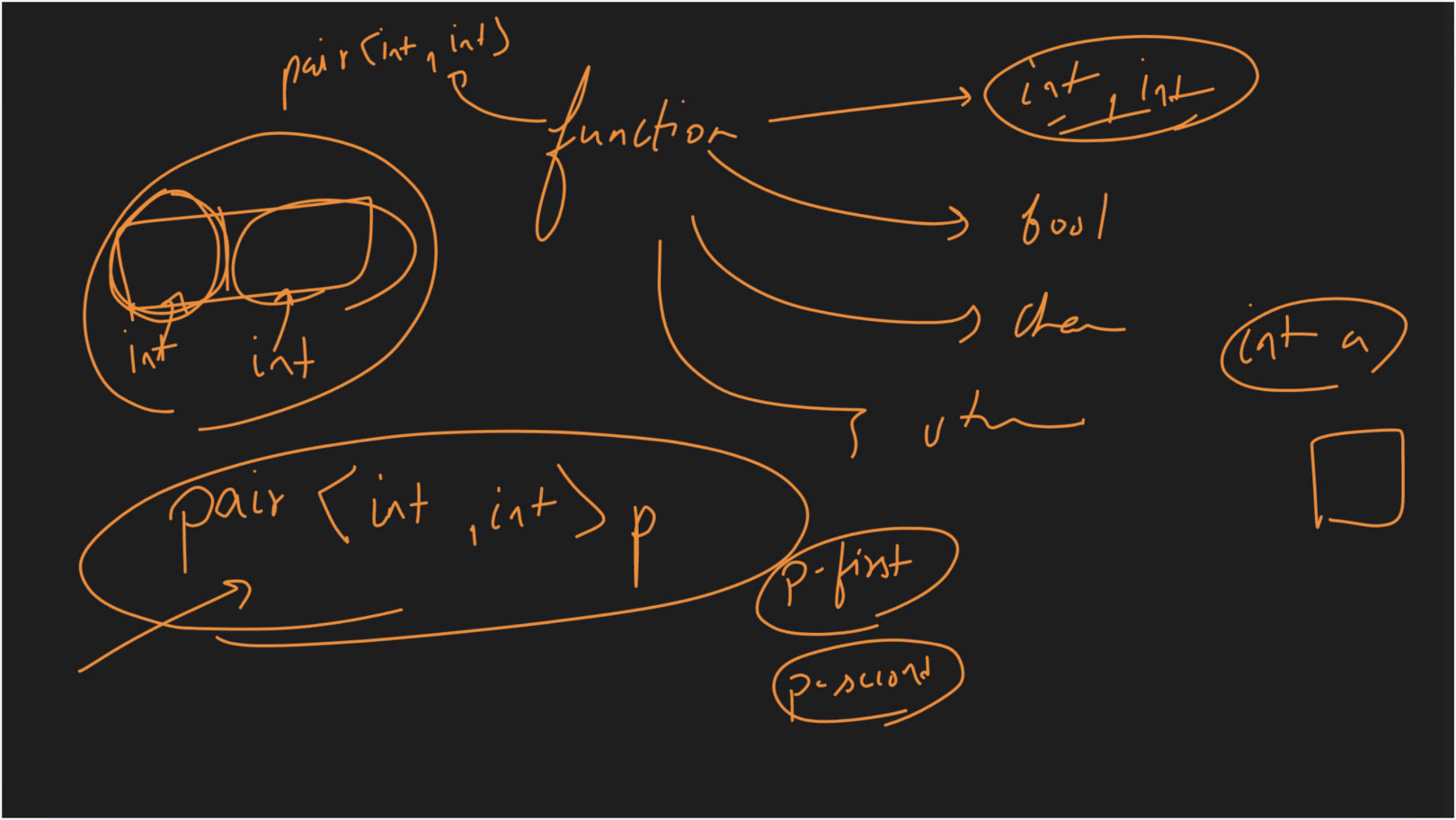
check if a pair that sums equal to target exist or not

Three Sum :

Imp Hassi Broo

Lag → reload


10 | 5 | 20 | 15 | 30
Lag - Sarcload
target = 35
i := 0 → n
j := 0 → n
arr[i] + arr[j] == target
10,10
10,5
10,20
10,15
10,30
5,10
5,5
5,20
5,15
5,30
20,10
20,5
20,20
20,15
20,30
15,10
15,5
15,20
15,15
15,30
30,10
30,5
30,10
30,15
30,30
5,30
20,15
15,20
30,5


pair<int, int>
function
int, int
bool
char
int a
int
int
pair<int, int> p
p.first
p.second

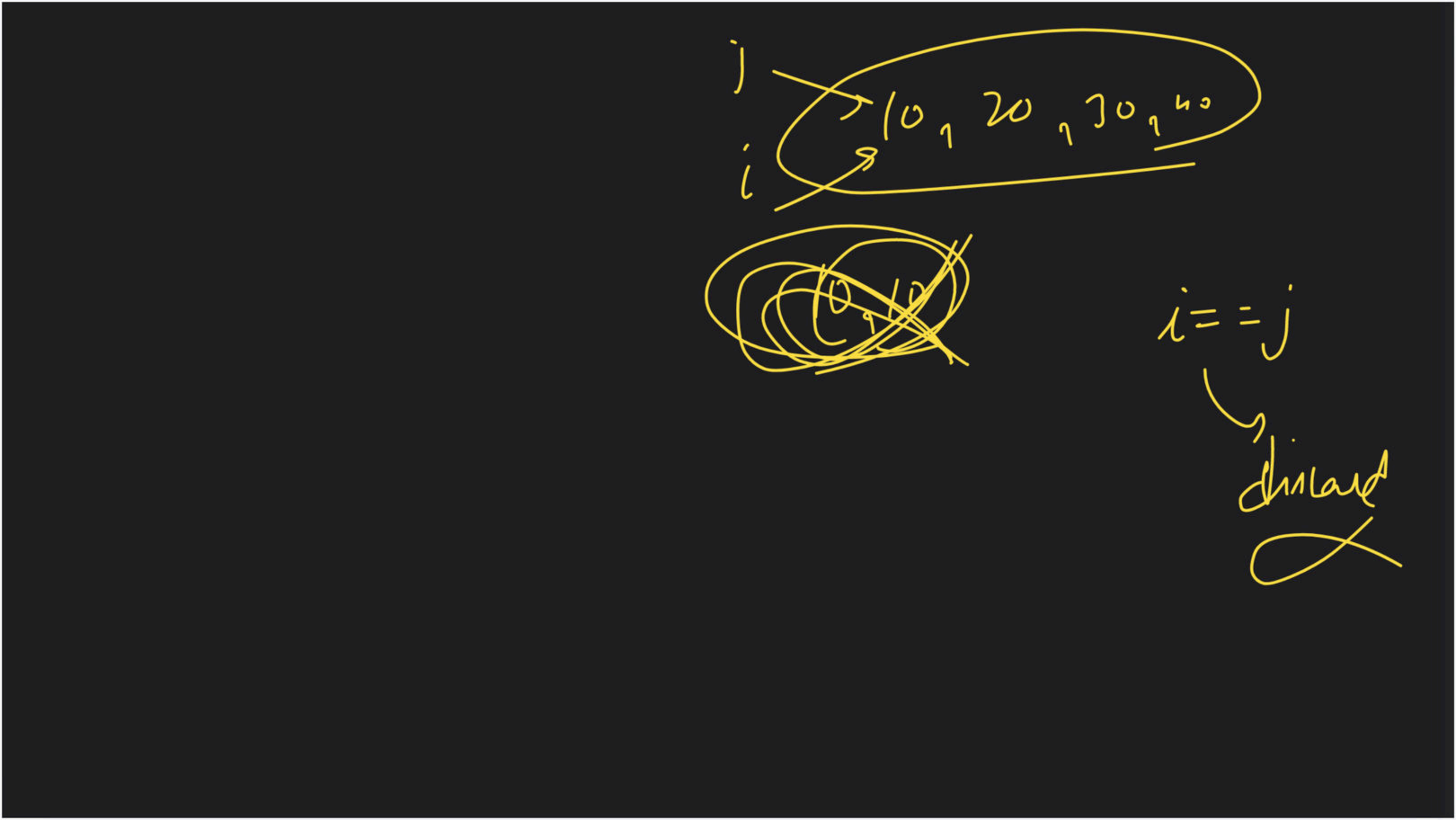
declare → pair<int, int> p;
↳ variable name

initialize → pair<int, int> p = make_pair(10, 20);

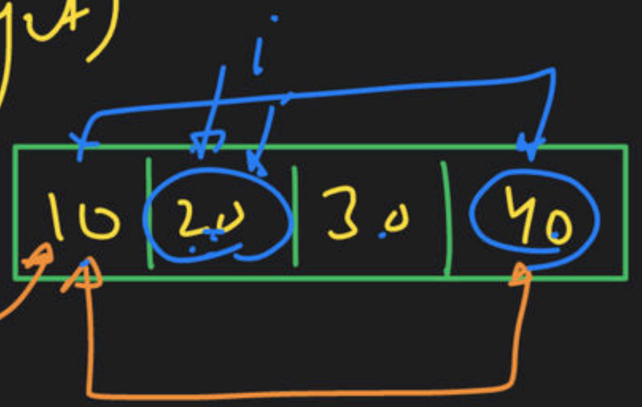
100	200

access → p.first = 100

p.second = 200


j
i
10, 20, 30, 40
i == j
Diwali

```
pair<int,int> Solve(int arr[], int n, int target)
{
    pair<int,int> p = make_pair(-1,-1);
    for(i=0 ⟶ <n)
    {
        for(j=0 ⟶ <n)
        {
            if(arr[i] + arr[j] == target)
            {
                p.first = arr[i];
                p.second = arr[j];
                return p;
            }
        }
    }
    return p;
}
```



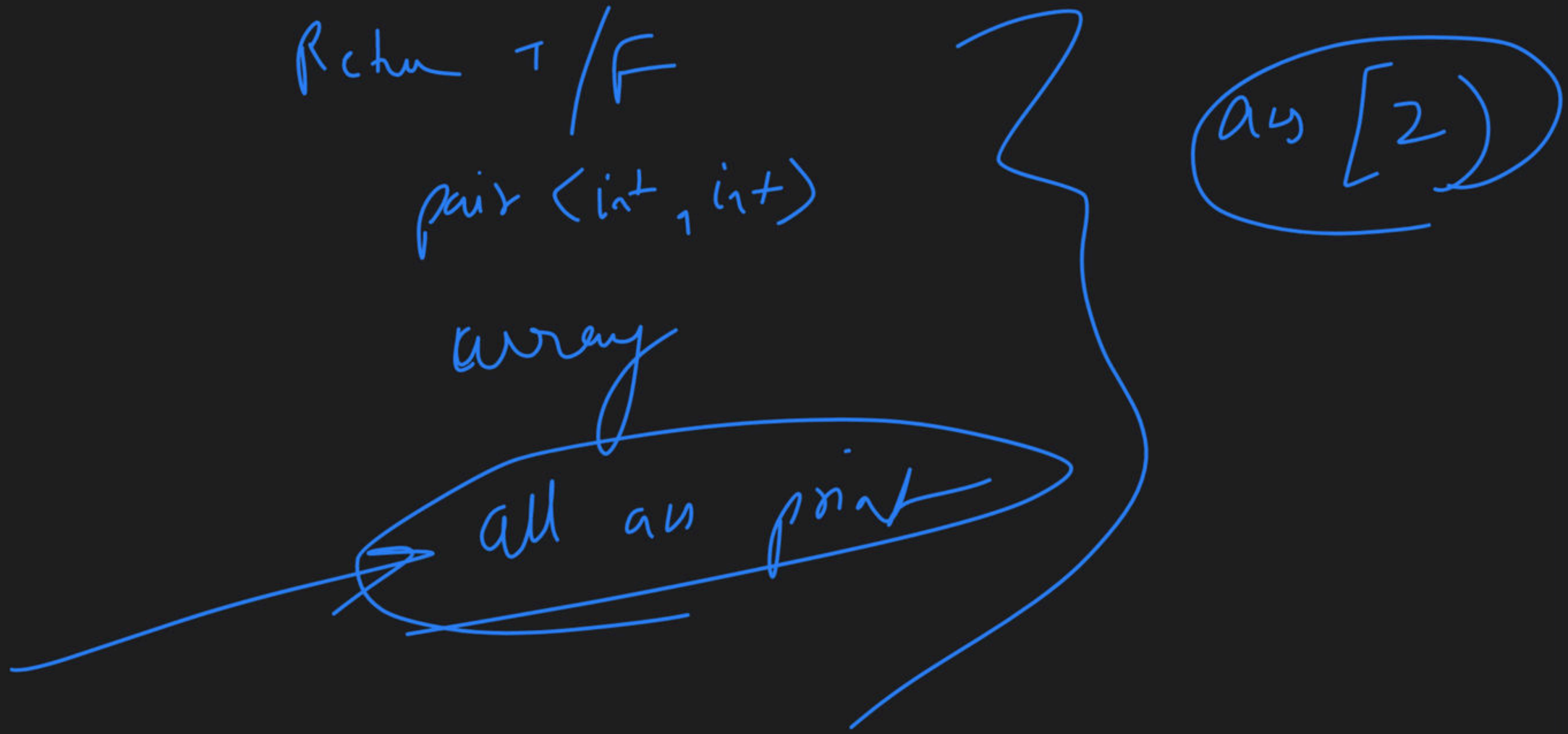
return

20 | 40

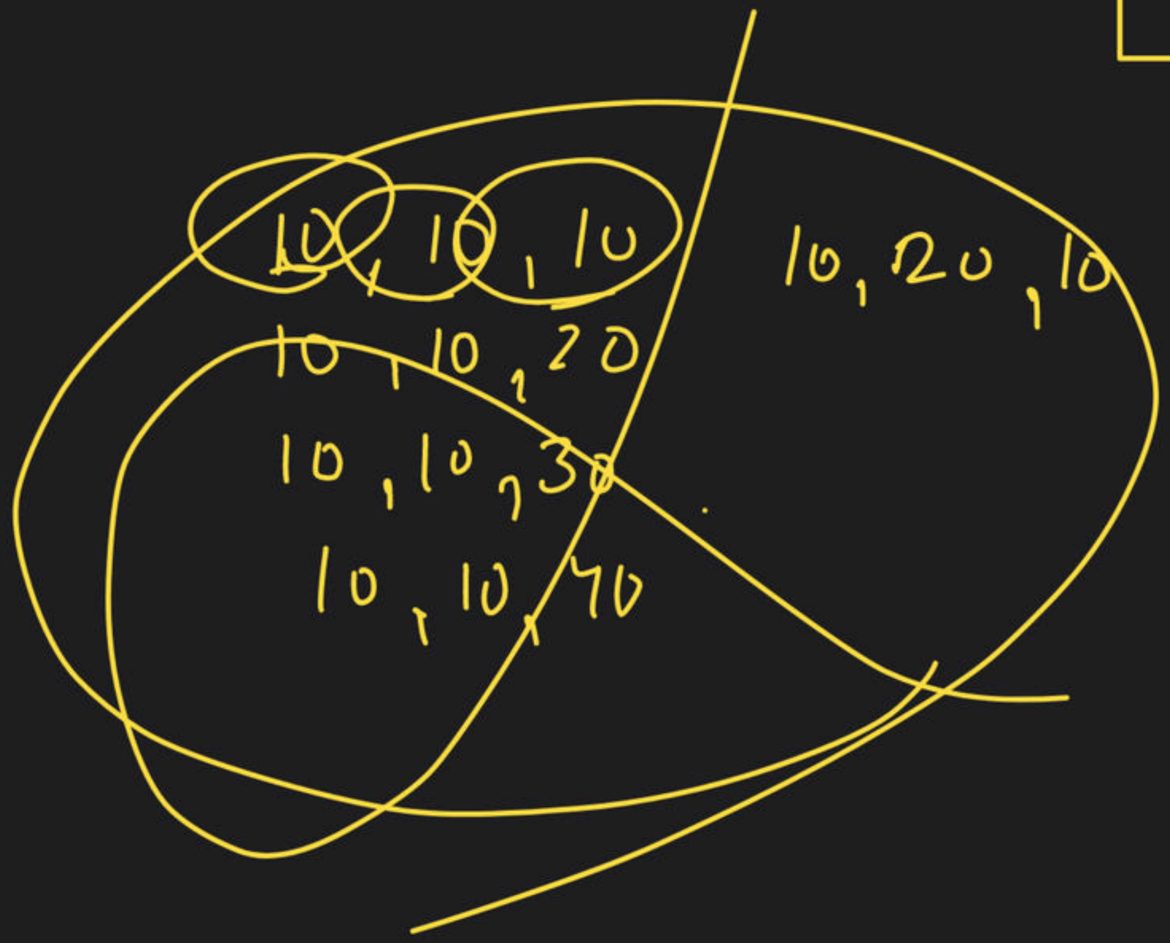
i

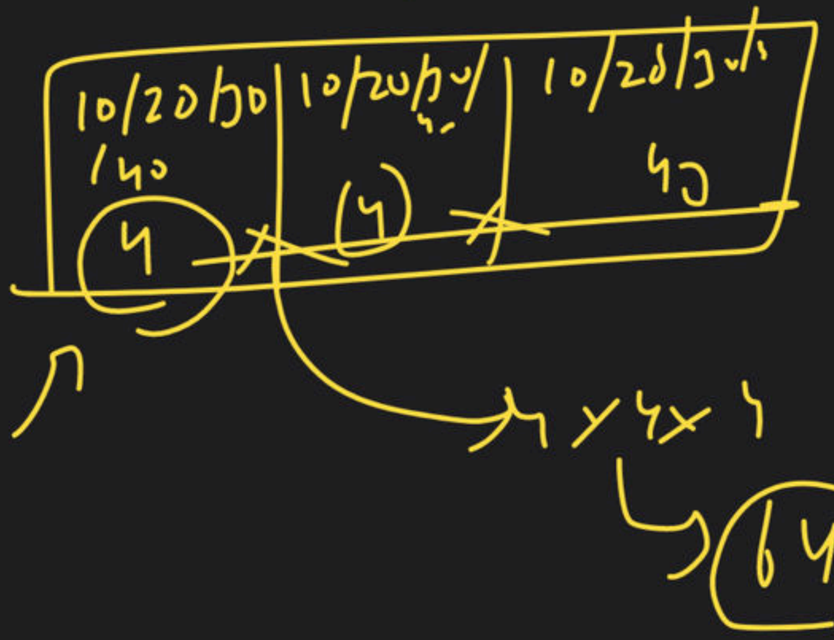
S.C → O(1)

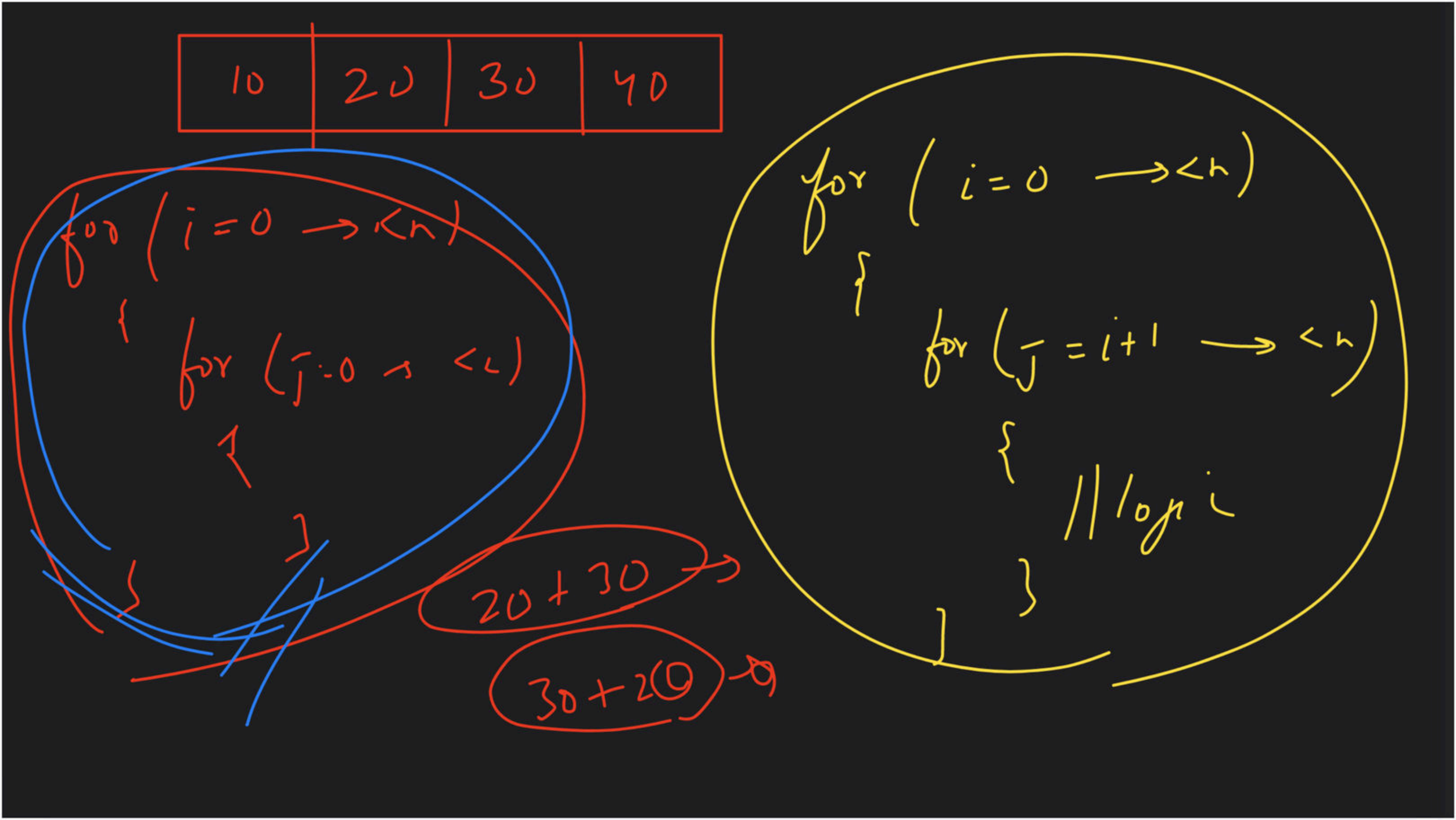
$O(n^2)$ → T.C

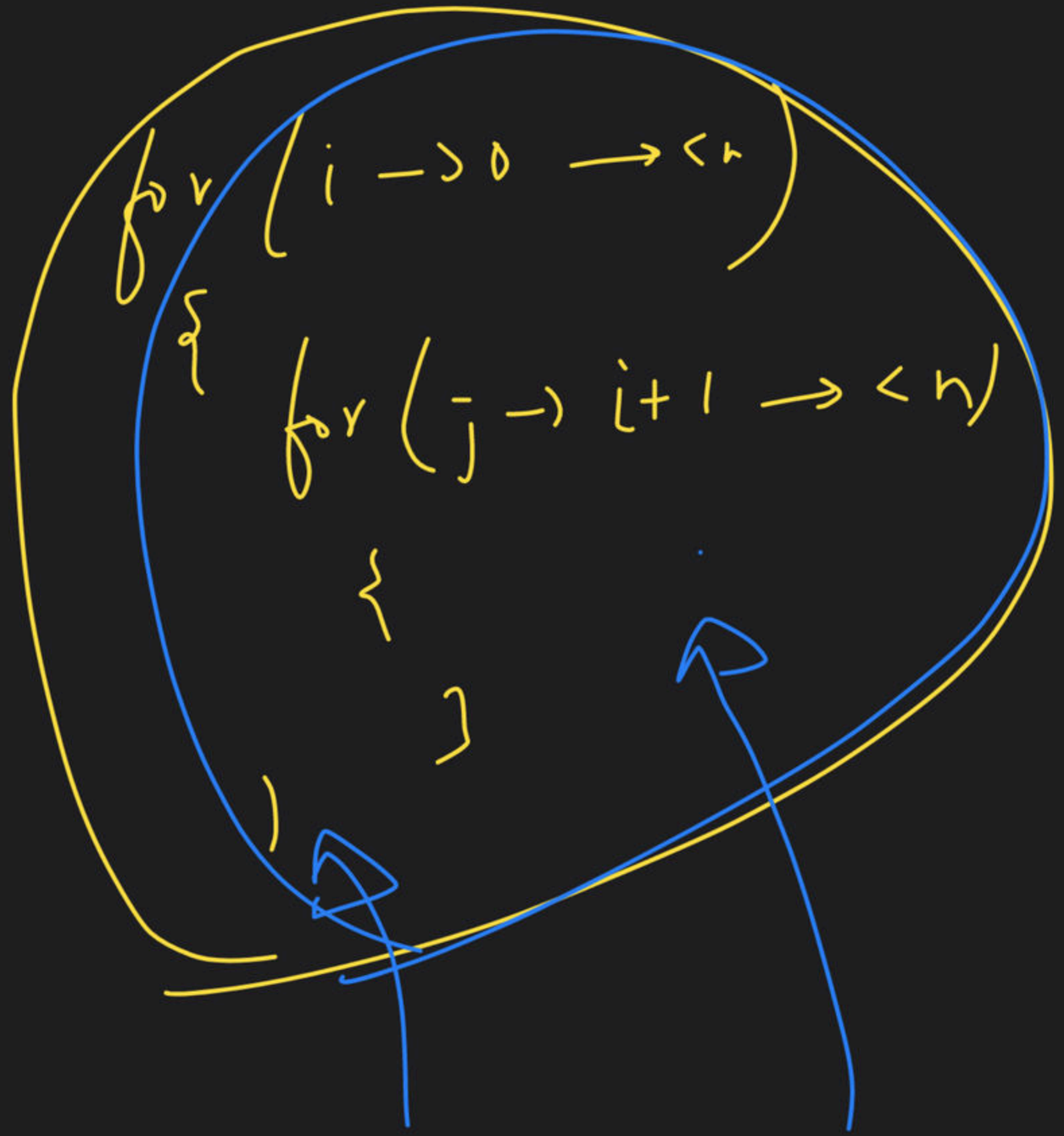

Return T/F
pair (int, int)
array
all as pair
as [2]

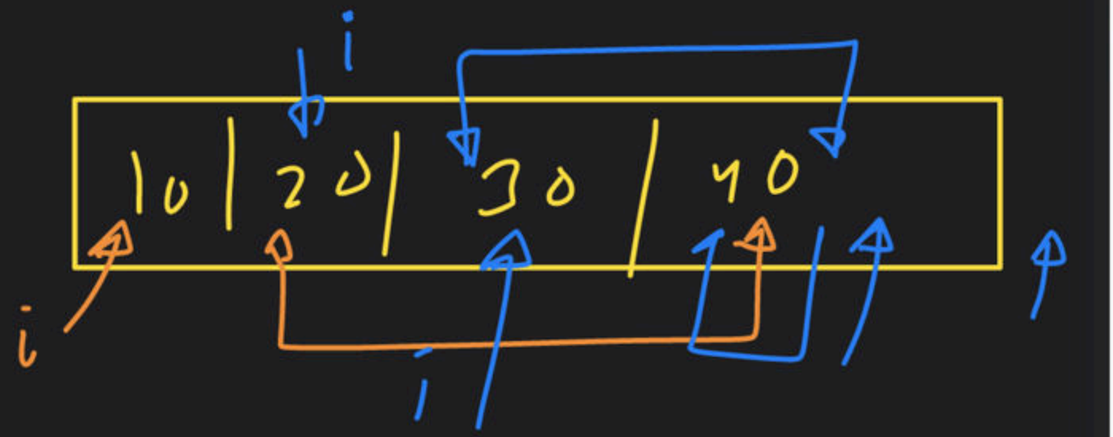
print all triplets





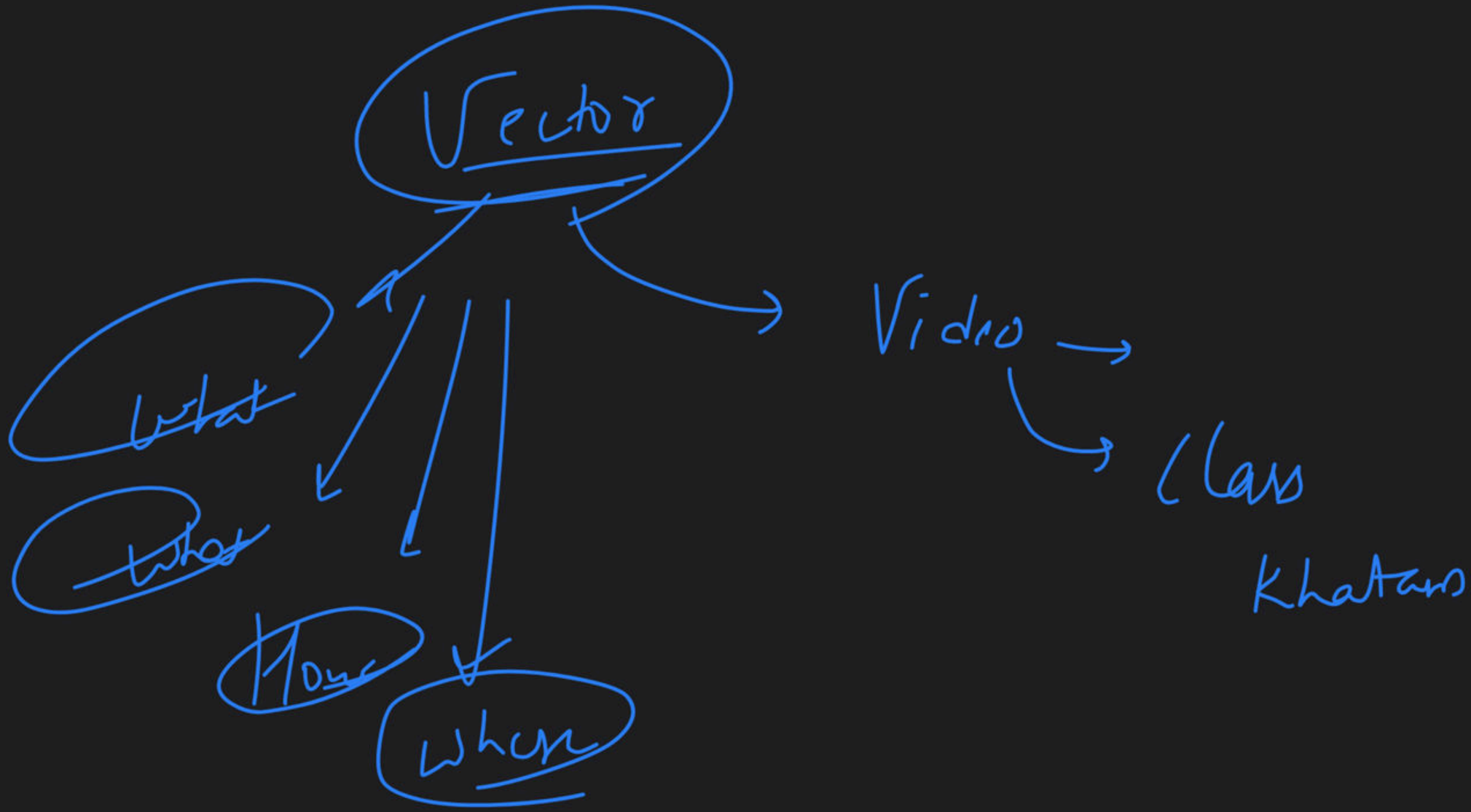

10
20
30
40
for (i=0 → <n)
{
for (j=0 → <i)
{
}
}
20 + 30
30 + 20
for (i=0 → <n)
{
for (j=i+1 → <n)
{
// logic
}
}

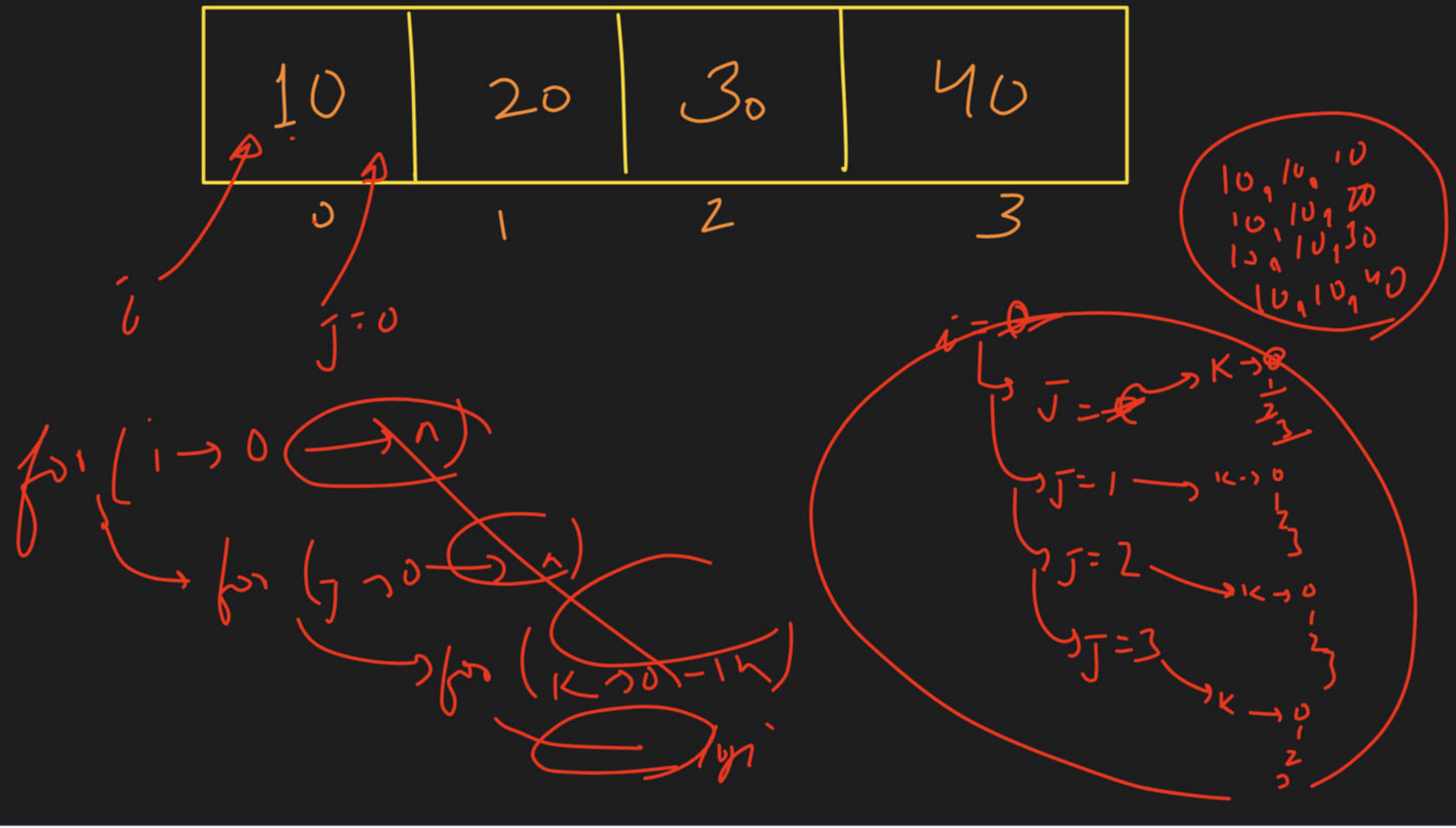




10,20	20,30	30,40
10,30	20,40	
10,40		

Unique relevant pair

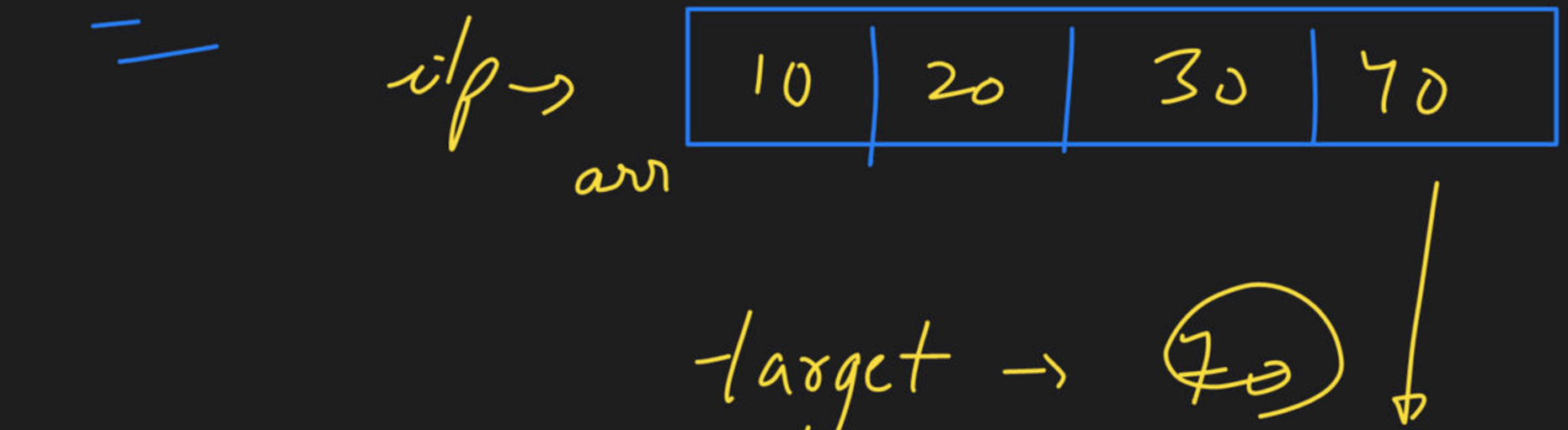

Vector
What
Why
How
When
Video
Class
Khatam


10
20
30
40
0
1
2
3
i
j=0
for (i→0 → n)
for (j→0 → n)
for (k→0 → n)
10 10 10
10 10 20
10 10 30
10 10 40
i=0
j=0 → k
j=1 → k→0
j=2 → k→0
j=3 → k→0

10	20	30	40

10	20	30
10	20	40
10	30	40
20	30	40

Three Sum



find a triplet

→ sum → target

10 20 30 → 60

10 20 40 → 70

20 30 40 → 90

10 20 40

Shift an Array

Video Repeat Revision

1 element

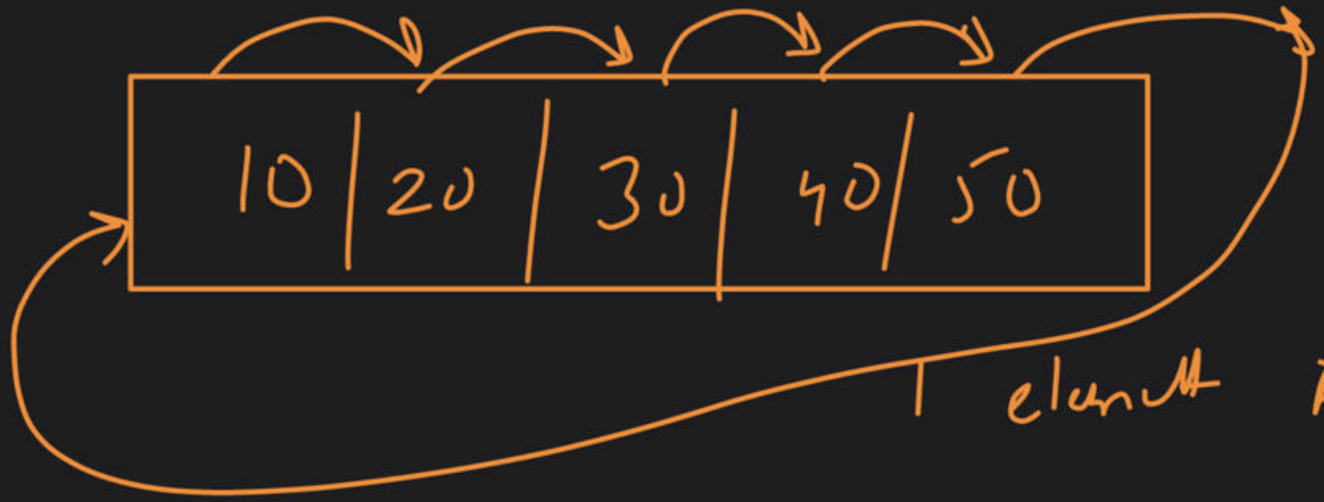
2 elements

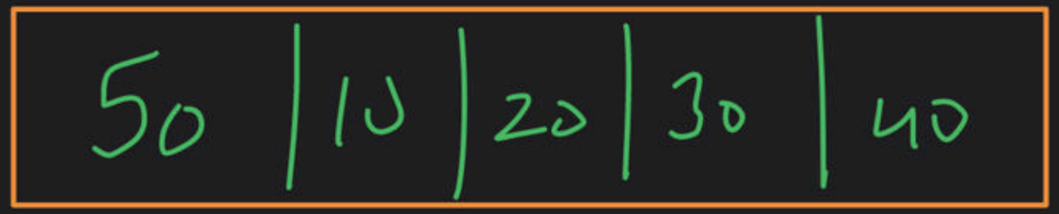
n elements

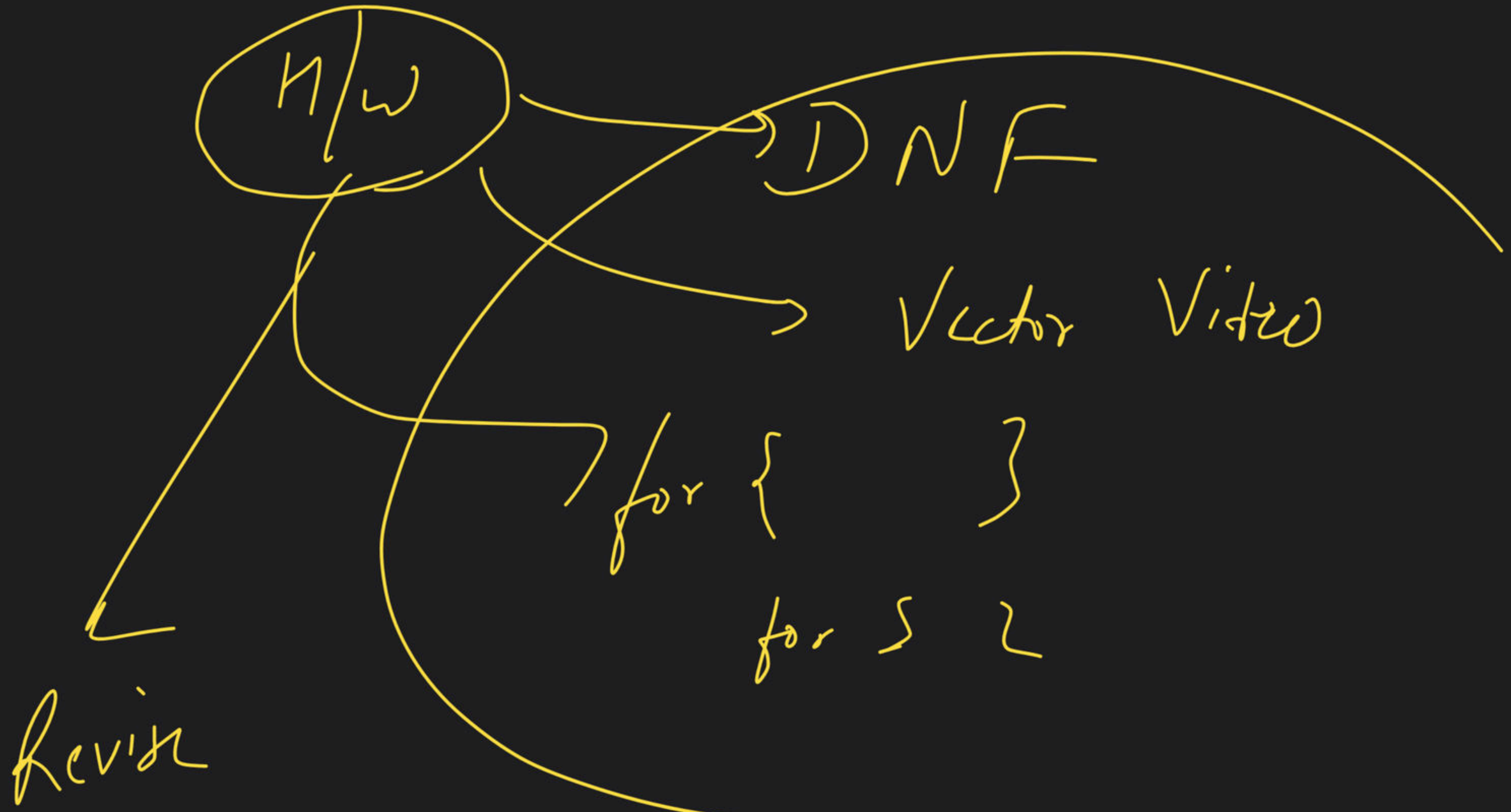
pattern

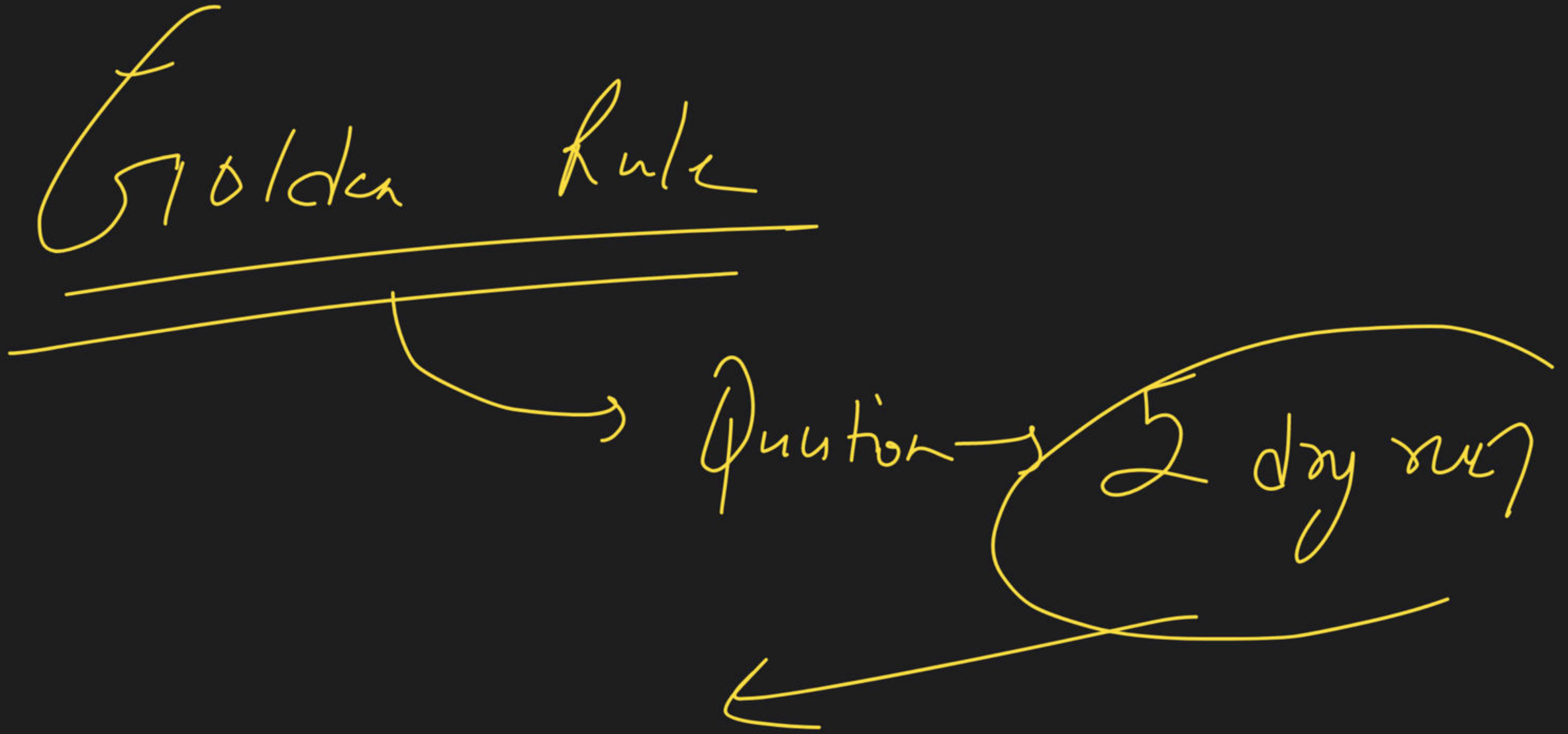
Swap

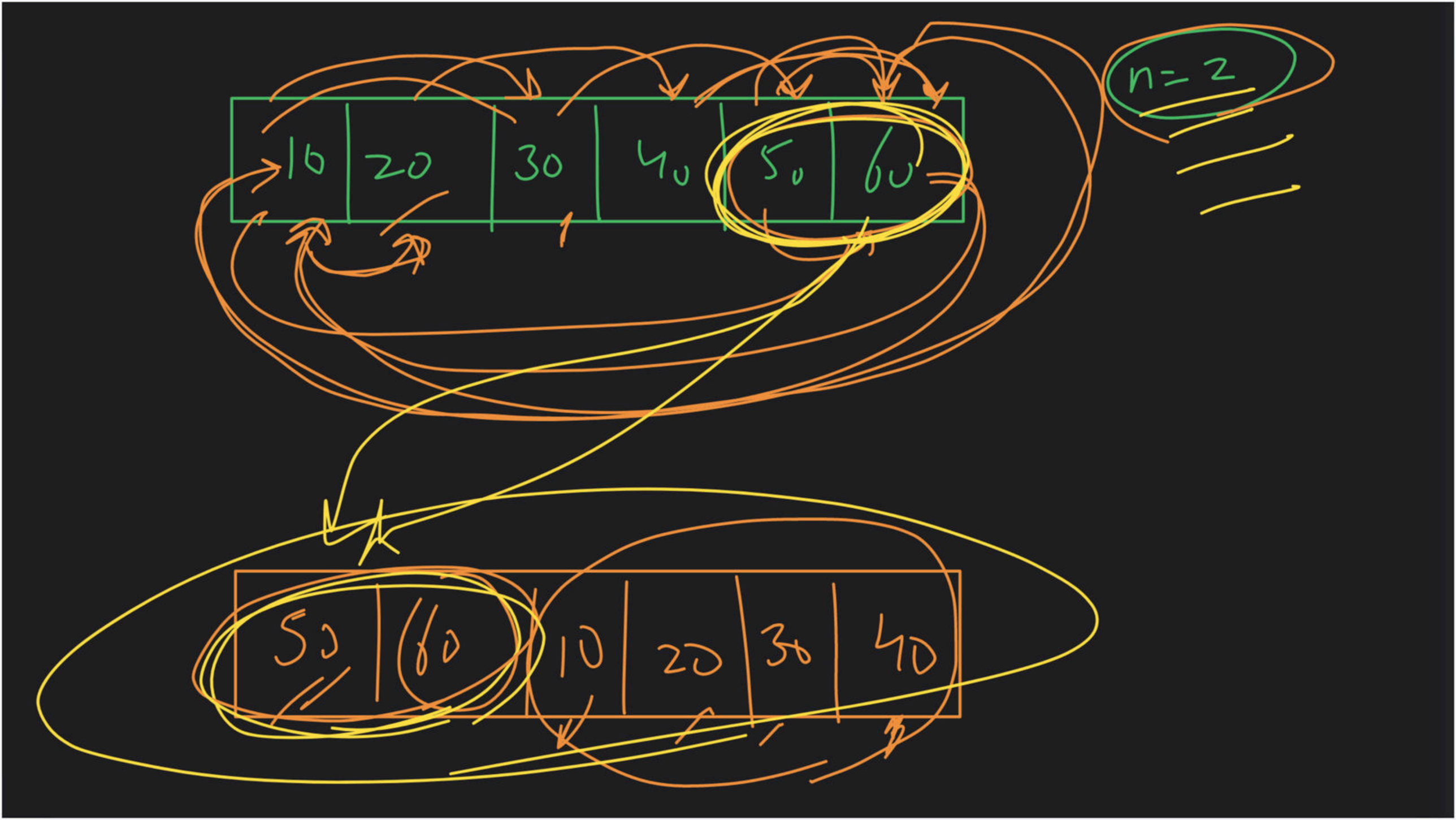
temp variable

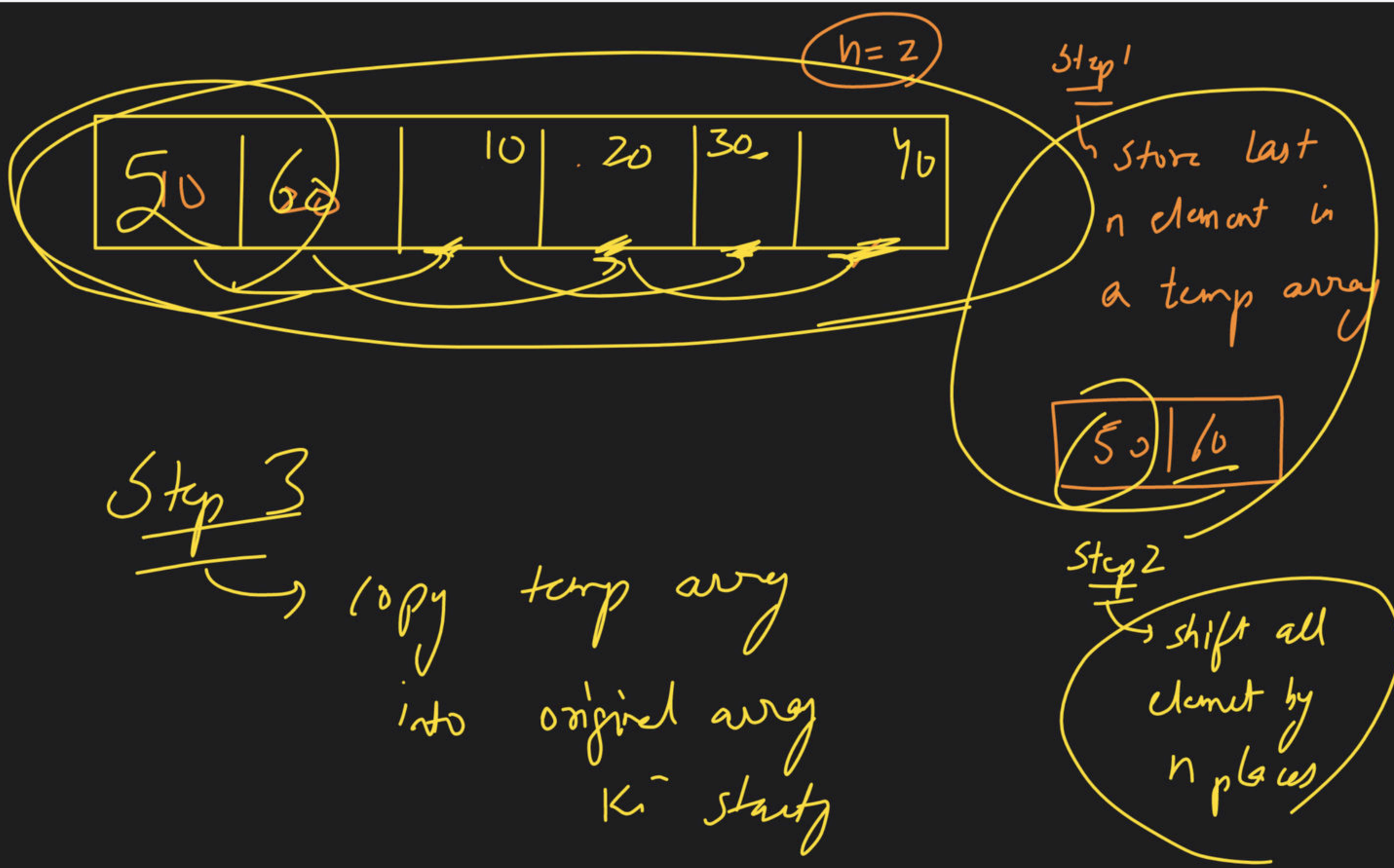


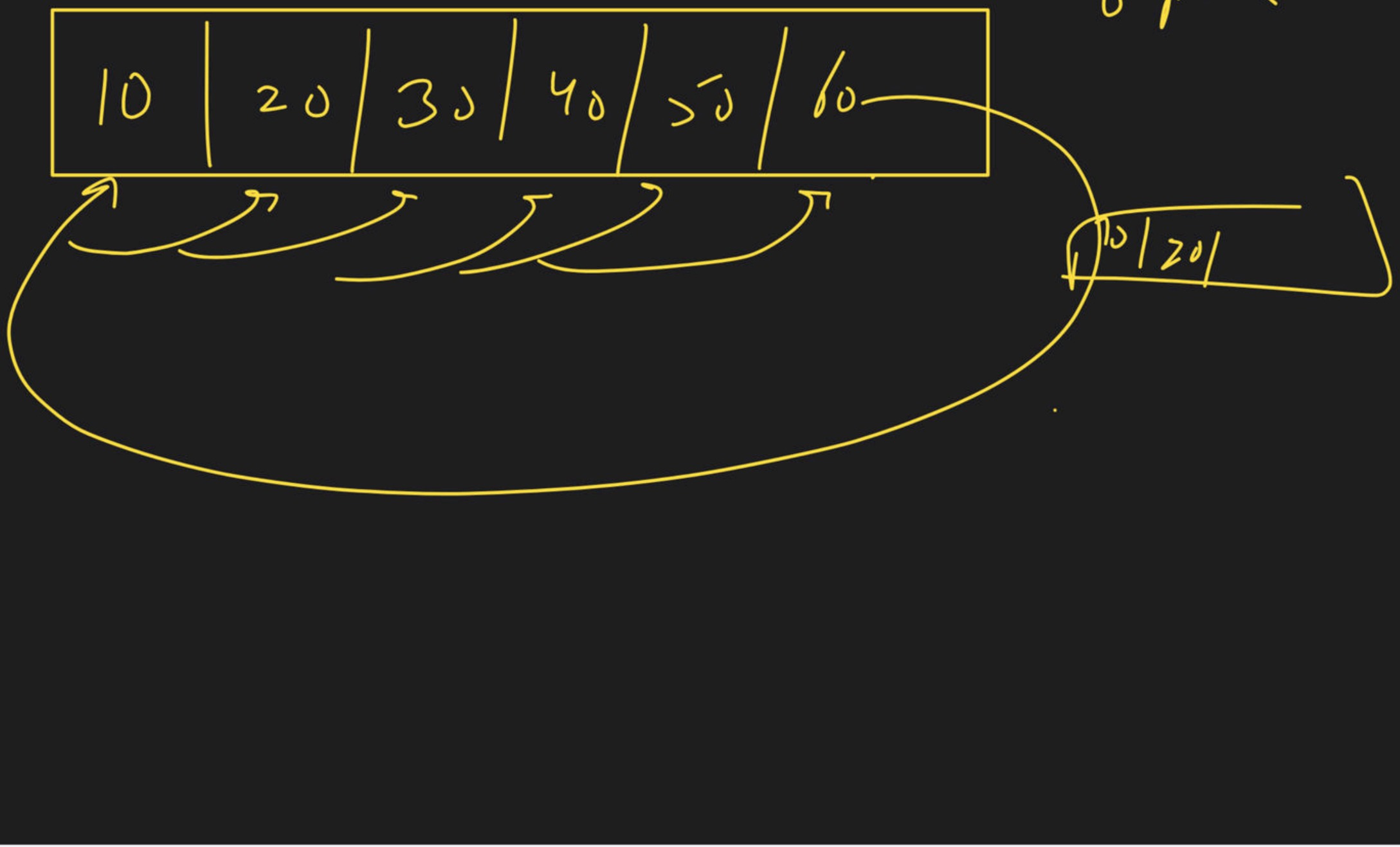


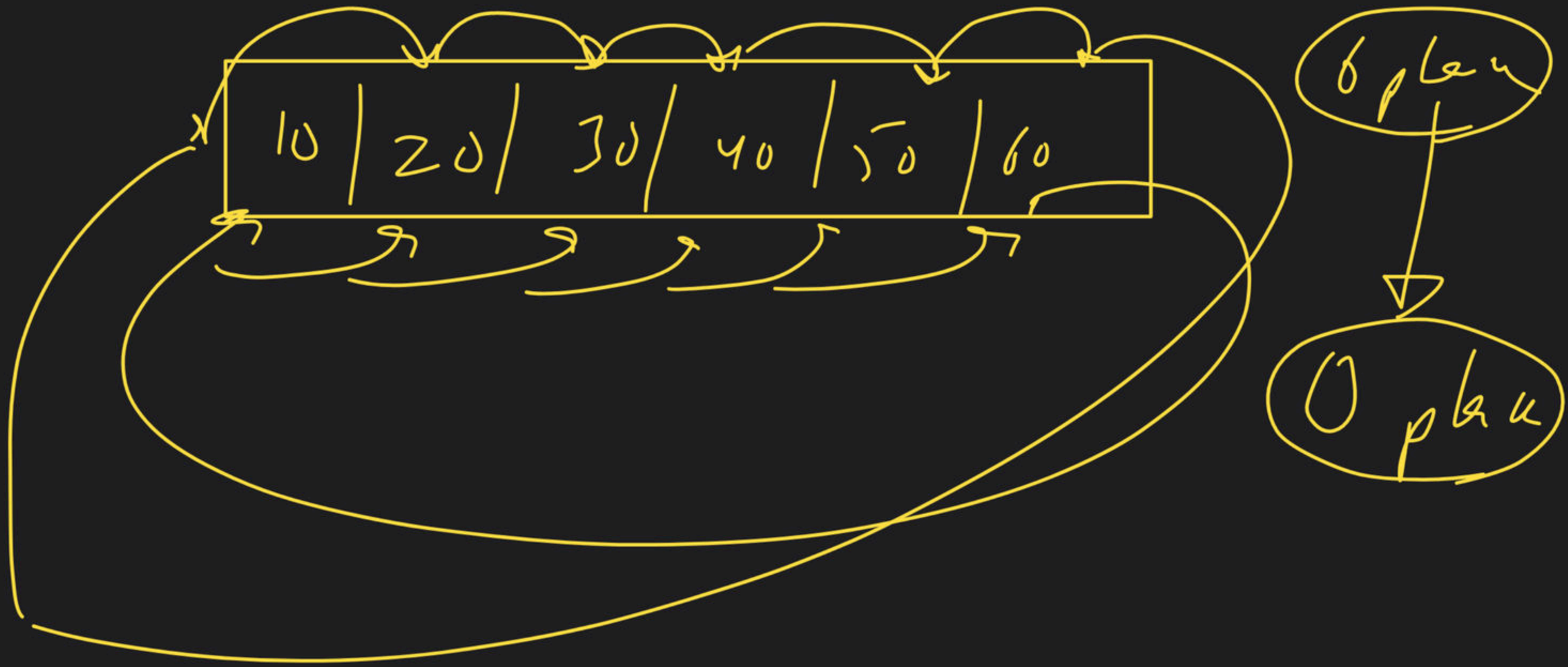

H/W
DNF
Vector Video
for { }
for ; ;
Revise

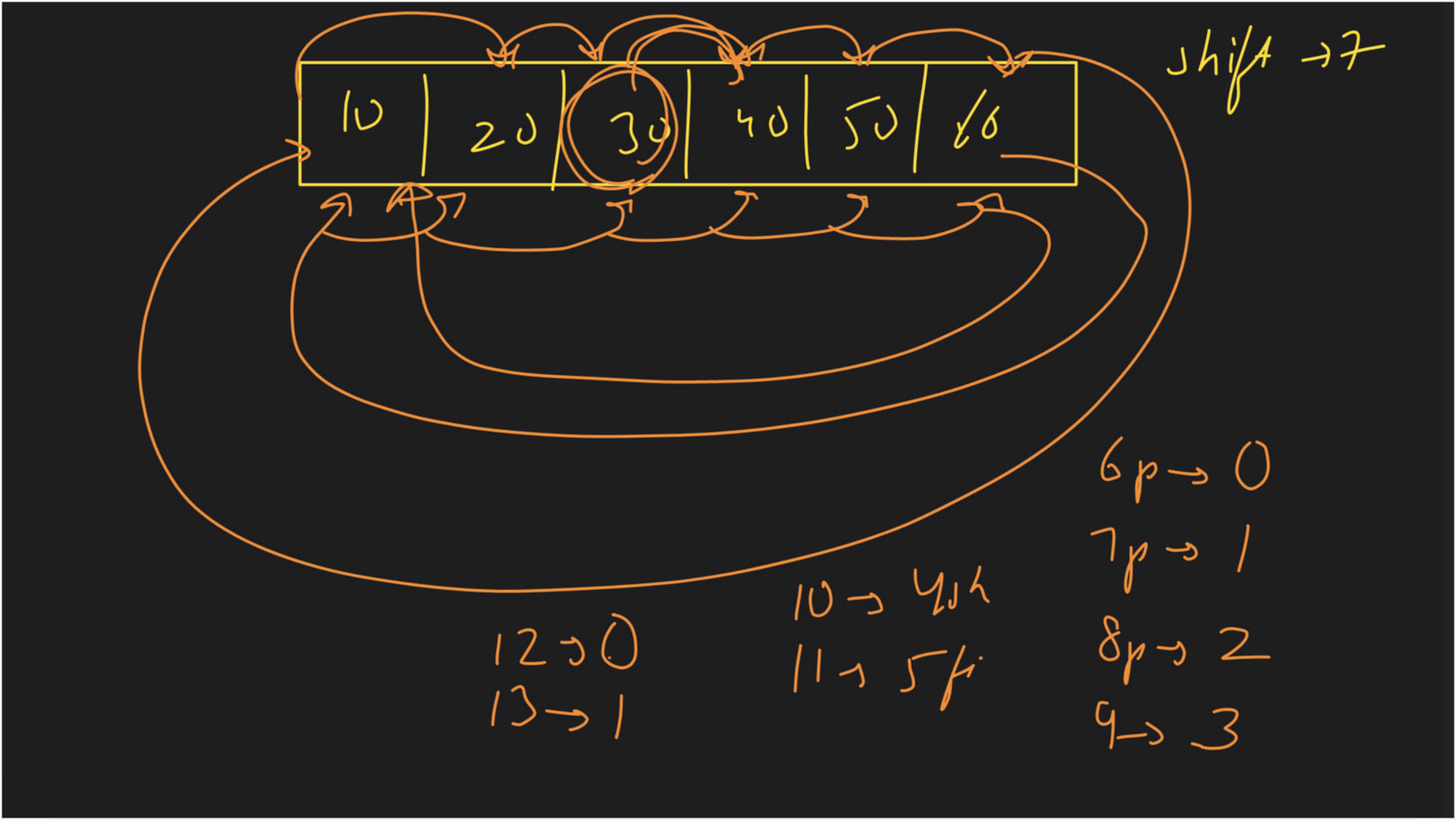

Golden Rule
Question
5 day run

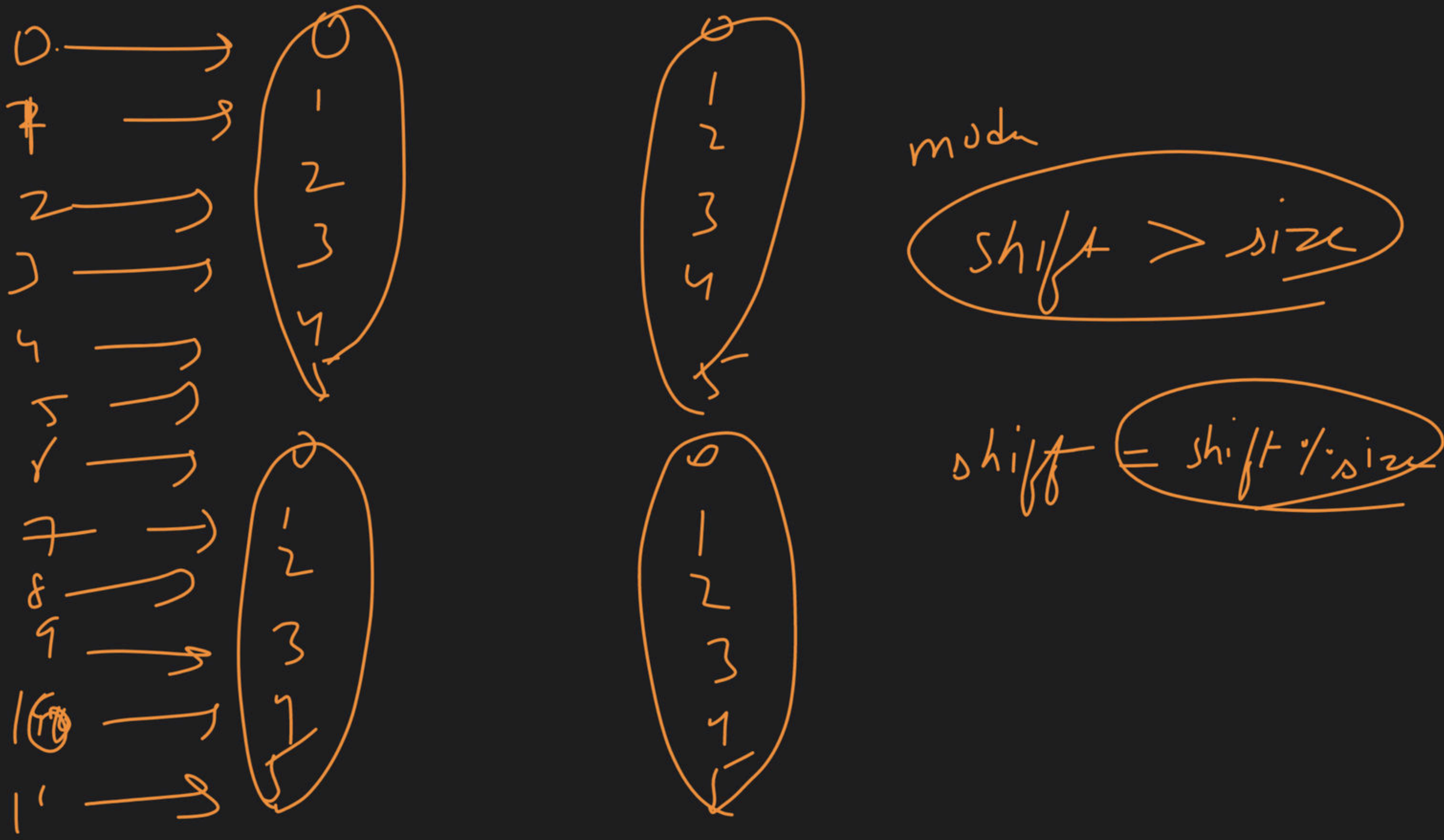

n = 2
10
20
30
40
50
60
50
60
10
20
30
40


n=2
50
60
10
20
30
40
Step 1
store last n element in a temp array
50
60
Step 2
shift all element by n places
Step 3
copy temp array into original array ki start


6 places
10
20
30
40
50
60
10
20


10
20
30
40
50
60
head
NULL


shift → 7
10
20
30
40
50
60
6p → 0
7p → 1
8p → 2
9 → 3
10 → ঘড়া
11 → ফড়ি
12 → 0
13 → 1


0
7
2
3
4
5
6
7
8
9
16
17
0
1
2
3
4
5
0
1
2
3
4
5
0
1
2
3
4
5
0
1
2
3
4
5
mod
shift > size
shift = shift % size

shift = 2

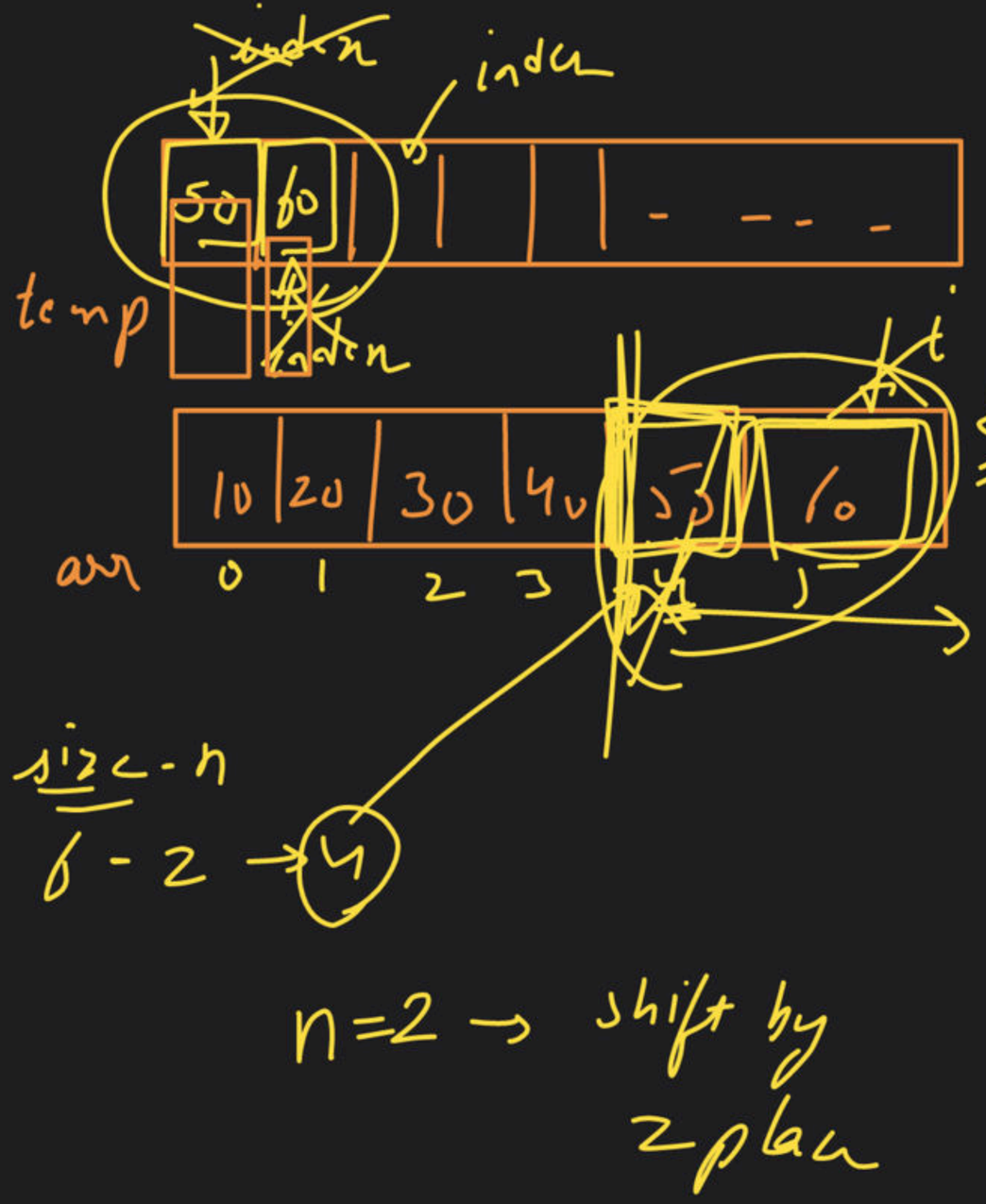
size = 6

<n

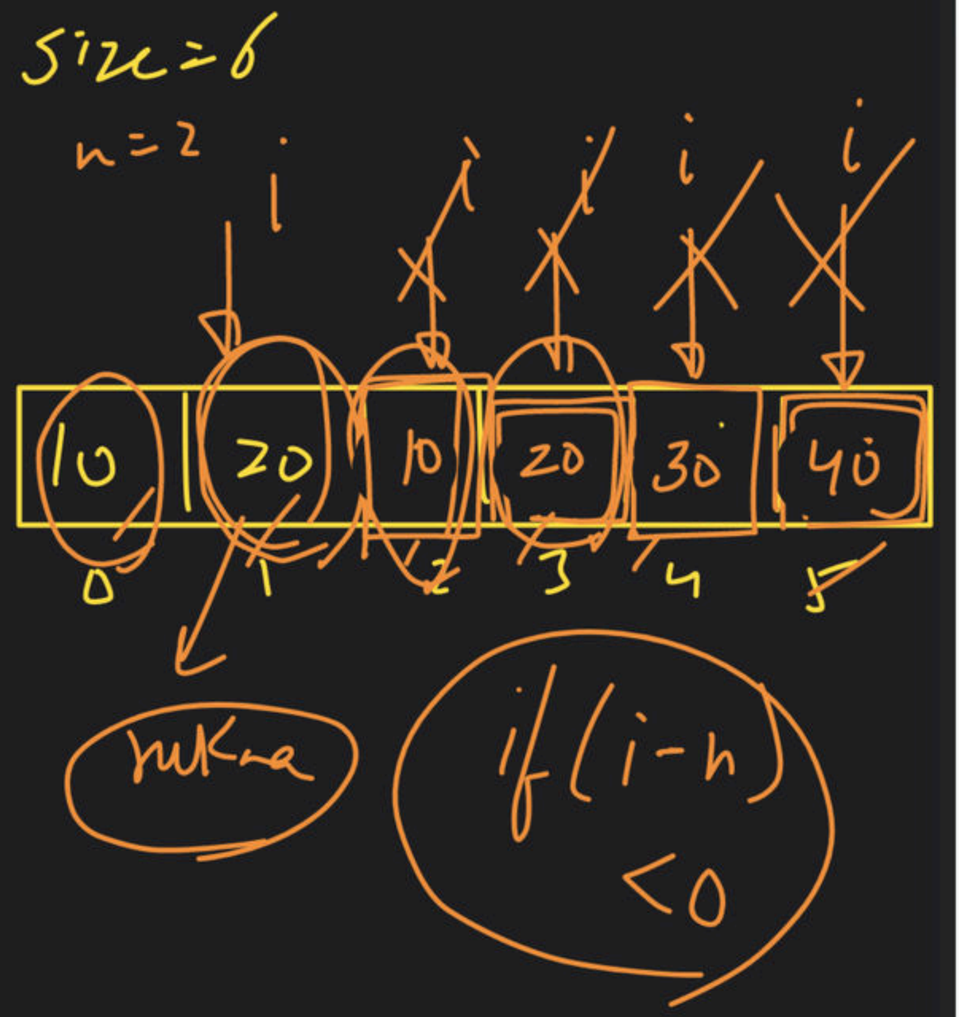
6 - 2 = 4

size - shift

```
int temp [10000];
int indexn = 0;
for (i = size - n; i<n; i++)
{
    temp [indexn] = arr[i];
    indexn++;
}
```



```
for (i = size-1 ; i>=0; i--)
{
    arr[i] = arr[i-n] ;
}
```



```
for (int i=0; i<=n; i++)
{
    arr[i] = temp[i]
}
```