Thursday, 14 October 2021 8:33 PM

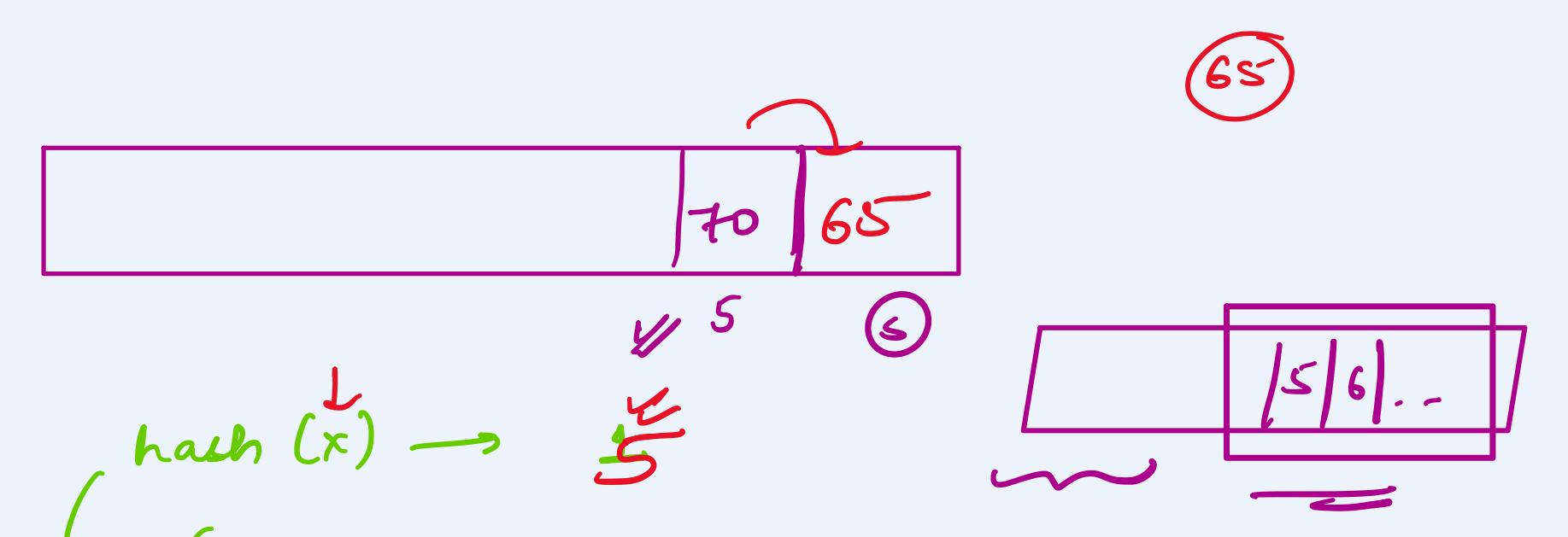
Collision Resolution Technique

→ **Open Addressing**

→ Linear Probing

key ↓ Hash Function ↓ index → (5)

(65)



hash(x) → 5

- (hash(x) + 1) % size
- (hash(x) + 2) % size
- (hash(x) + 3) % size

→ **Quadratic probing**

i^2 slot

hash(x)

- $(hash(x) + 1^2)$ % s
- $(hash(x) + 2^2)$ % 5
- $(hash(x) + 3^2)$ % 5

→ **Double hashing**

hash(x)

Introduce hash2(x)

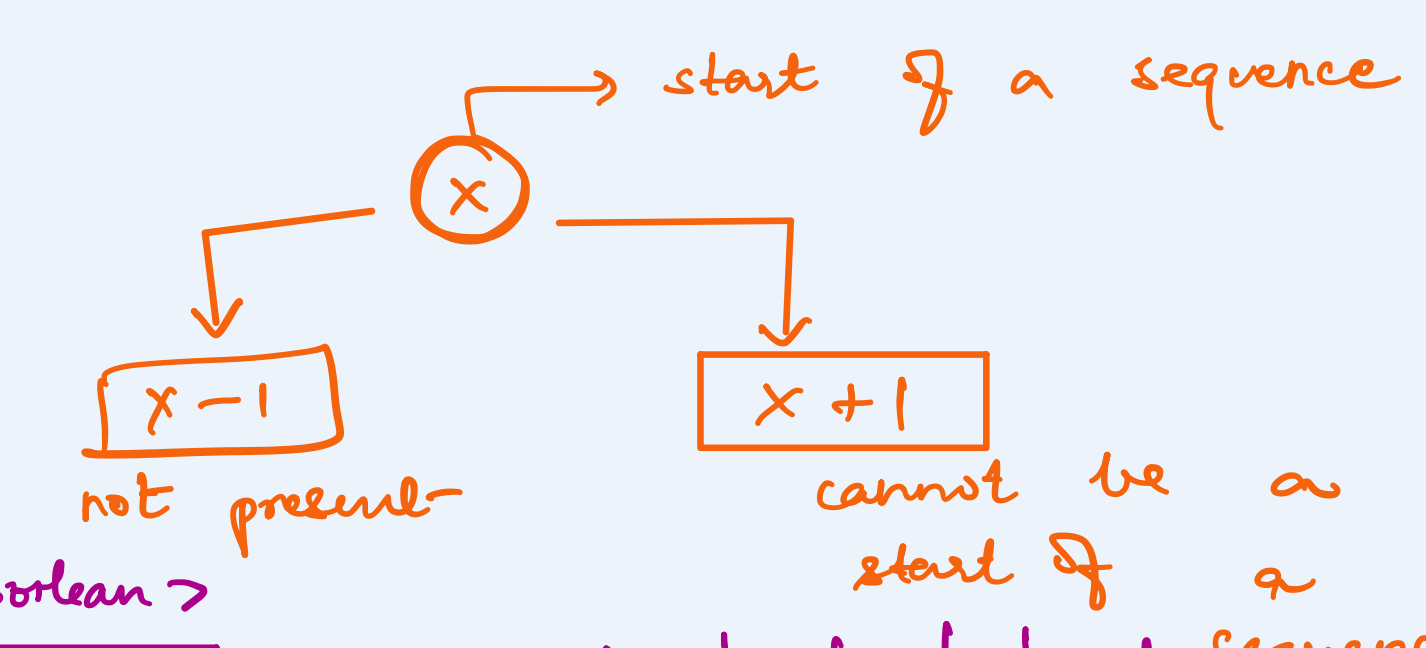
hash(x)

- (hash(x) + 1 * hash2(x)) % s
- (hash(x) + 2 * hash2(x)) % s
- (hash(x) + 3 * hash3(x)) % s

Q Longest Consecutive Sequence

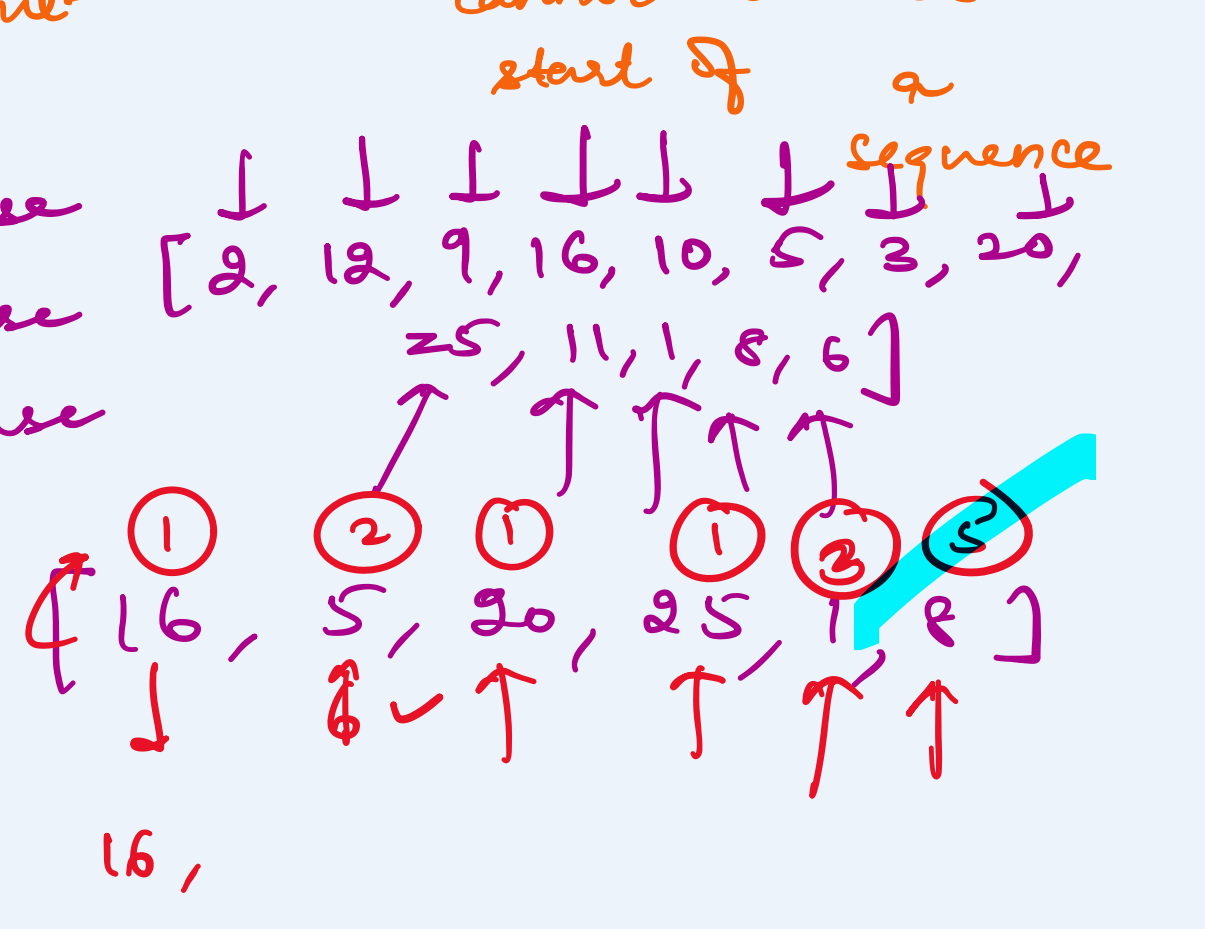
arr = [2, 12, 9, 16, 10, 5, 3, 20, 25, 11, 1, 8, 6]

ans → (5)



< Integer, Boolean >

2	~~true~~	false
12	~~true~~	false
9	~~true~~	false
16	true	
10	false	
5	true	
3	false	
20	true	
25	true	
11	false	
1	true	
8	true	
6	false	



Q Given an integer array and a number k, return true / false if the given array can be divided into pairs such that sum of every pair is divisible by k.

Eg: arr = [9, 7, 5, 3]   k = 6

output → true

(9, 3) → 12 ✓   (7, 5) → 12 ✓

Eg: arr = [91, 74, 66, 48] , k = 10

output → false

74 + 66 → 140 ✓   (91 + 48) → ✗

(1) if (arr.length % 2) != 0 → false

Given two nums (a) and (b)

✓ a % k == x   2 ✓ b % k == k − x

then → (a + b) % k == 0

Proof: a % k == x, b % k == k − x → given

$(a+b)$ % k = ((a + b) % k) % k

= (a % k + b % k) % k

= (x + k − x) % k

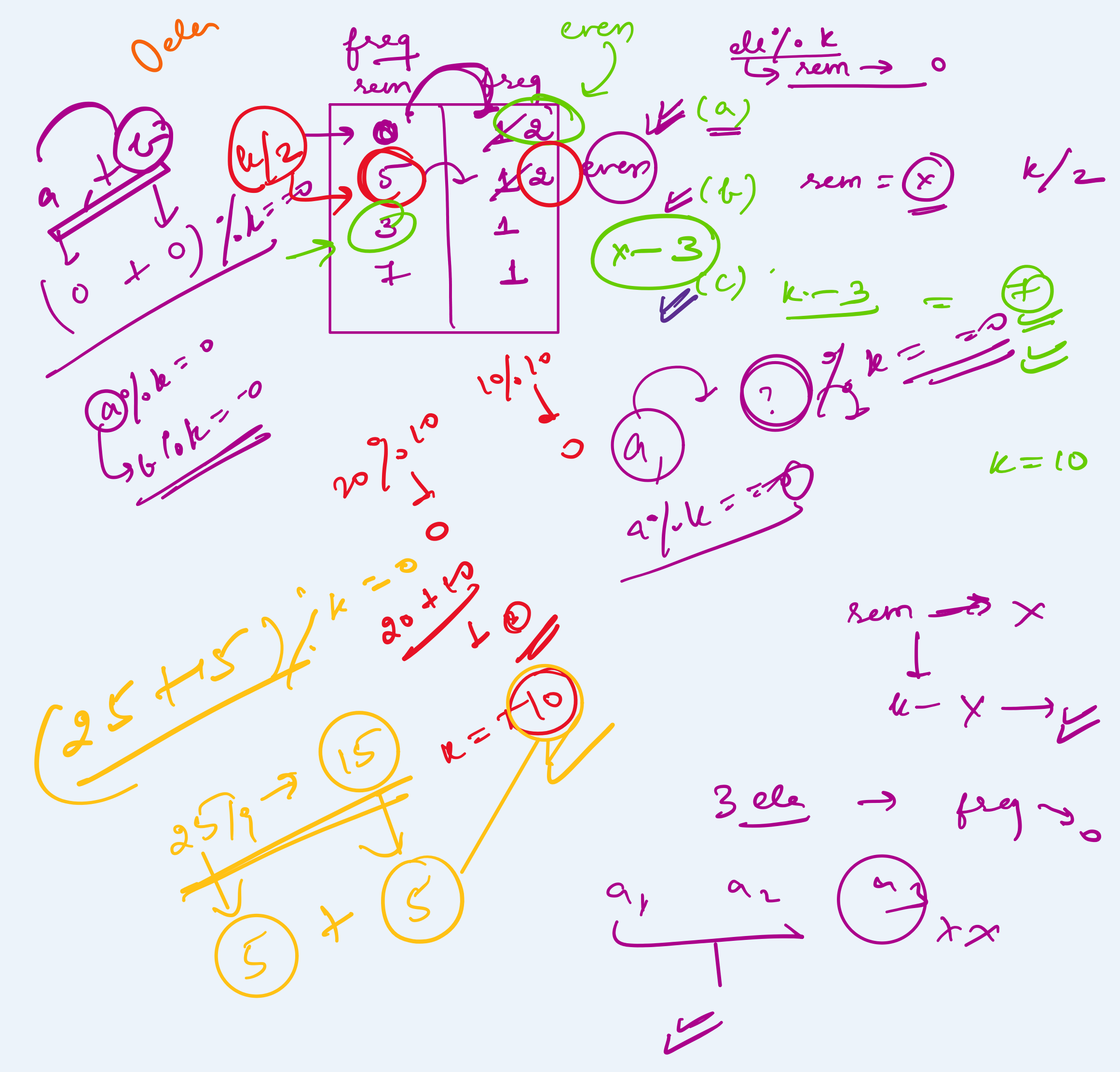
= k % k

= 0

(a + b) % k = 0

(a + b) is divisibly by (k).

arr → [20, 25, 10, 3, 15, 7] , k = 10

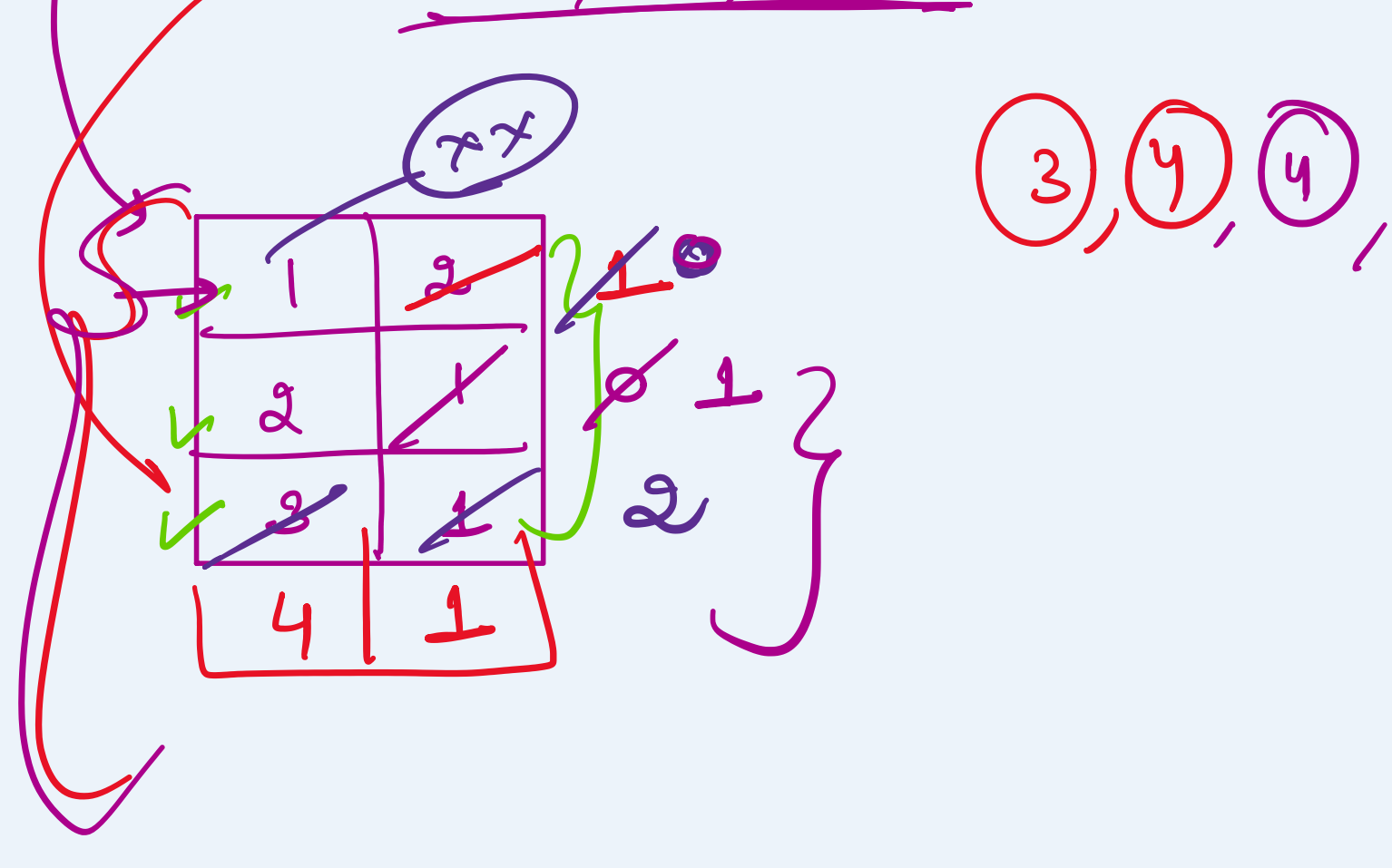


Q Given an integer of size (n) and integer (k), return the count distinct no. in all windows of size (k).

arr → [1, 2, 1, 3, 4, 2, 3]   k = 4

ans → 3, 4, 4, 3

(3), (4), (4), (3)



Q Find the length of the longest subarray with 0 sum.

arr → [15, −2, 2, −8, 1, 7, 10, 2, 3]

ans → (5)

arr → [1, 2, 3]

ans → 0