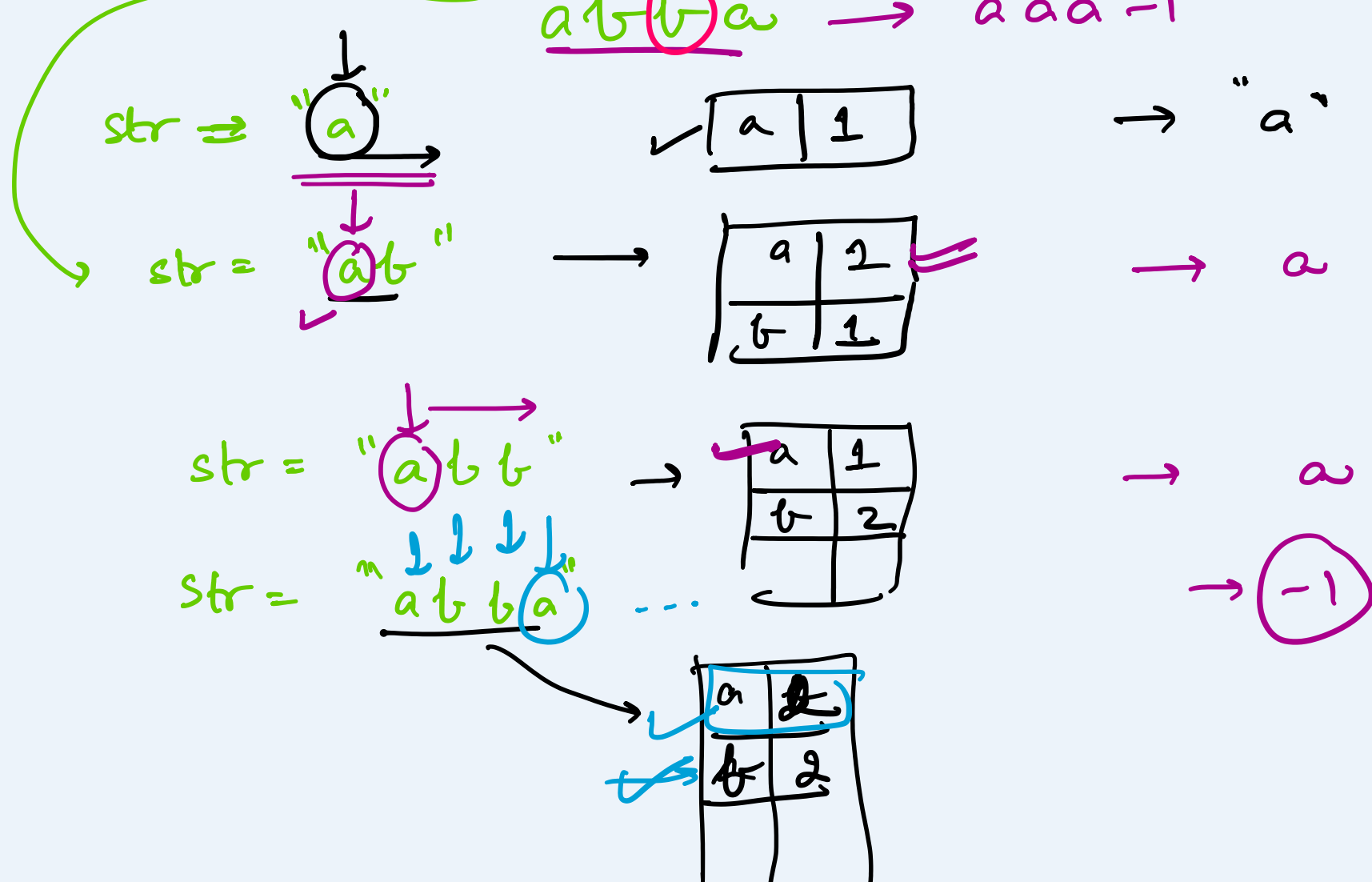


Q. Given a stream of characters, find the first non-repeating character from the stream.

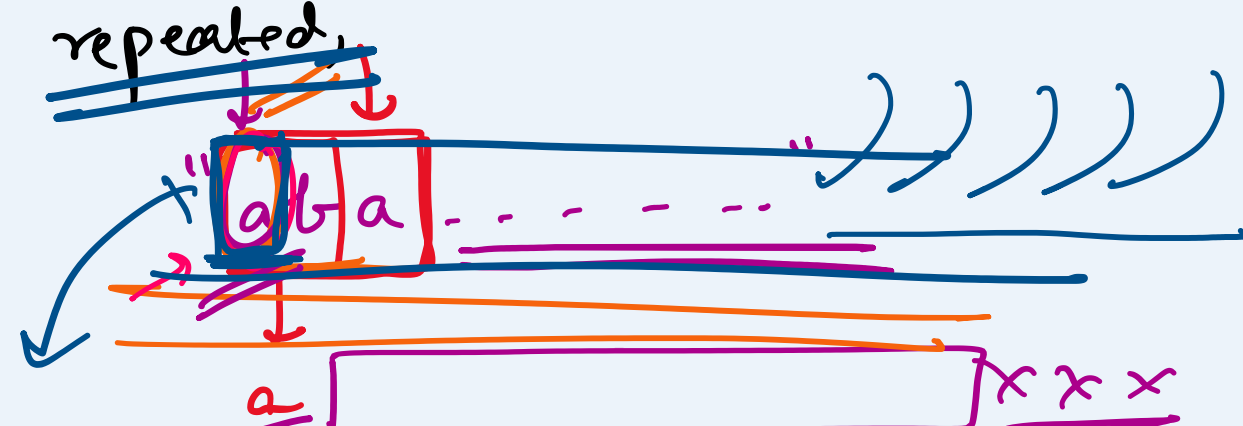
I → "aabc"
 O → "a-b-b"
 "a" → a
 "aa" → -1
 "aab" → b
 "aabc" → b

I → "abacac"
 → a a b b b b

Brute force approach



Intuition: let's say char 'a' has been repeated.



a b a c c c b

a a b b b b

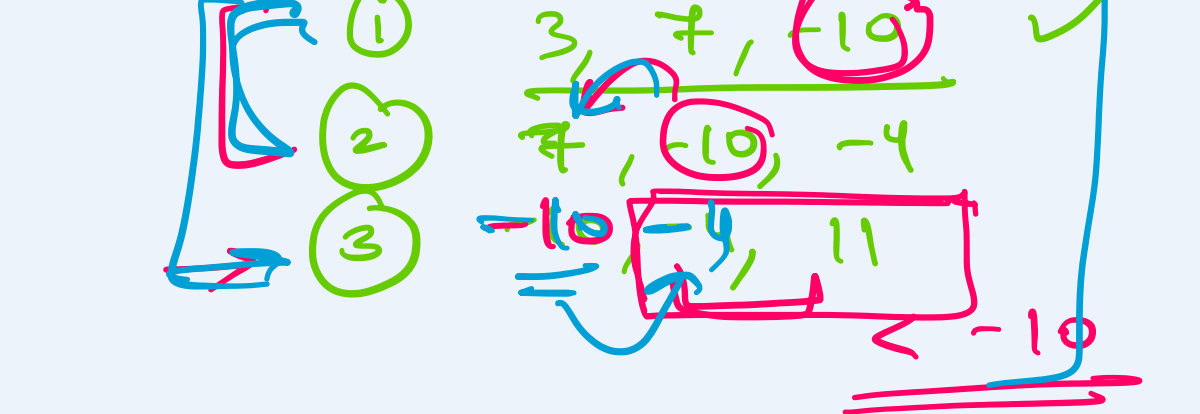
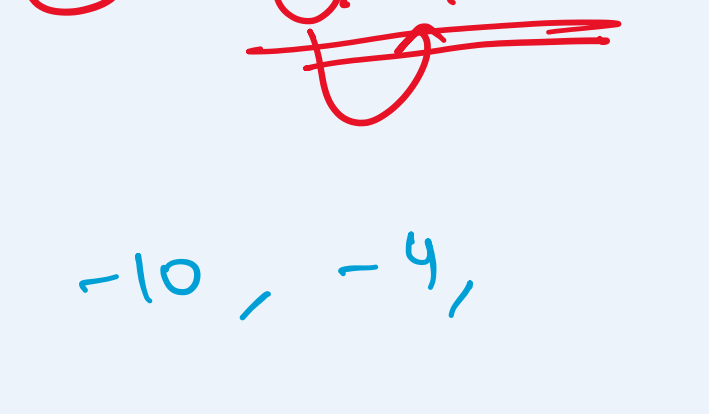
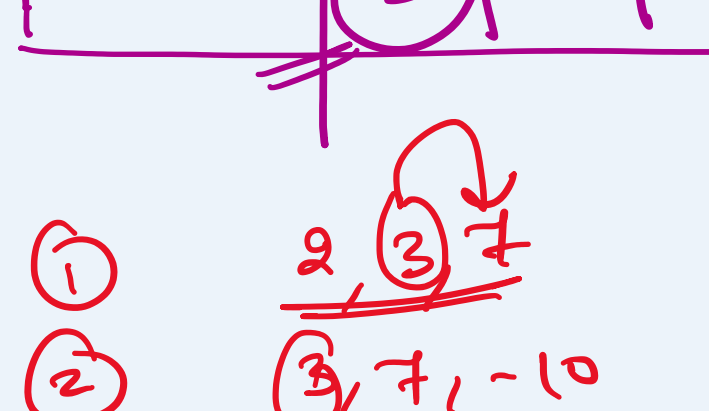
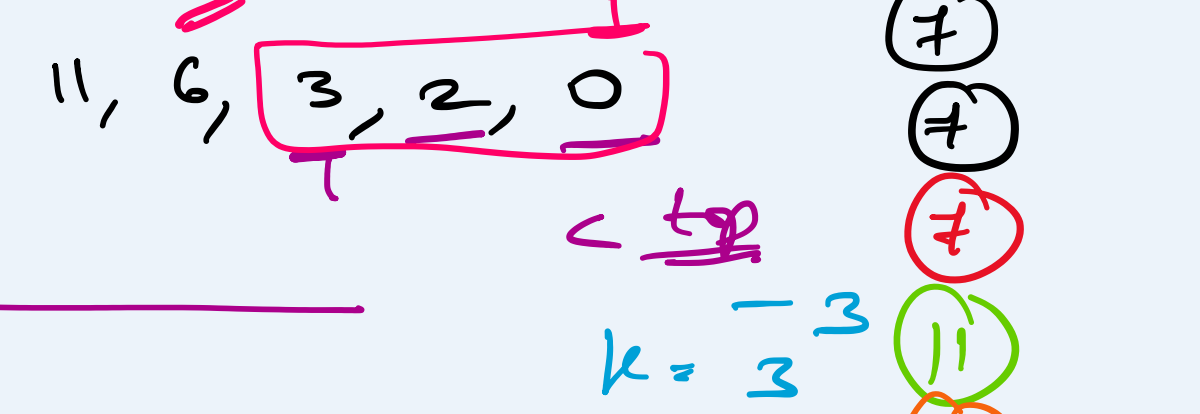
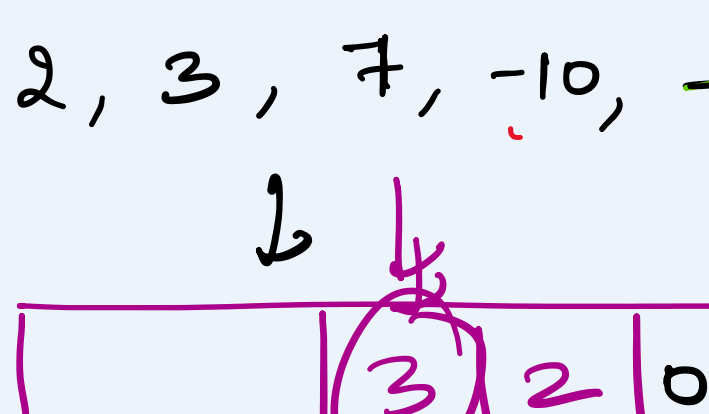
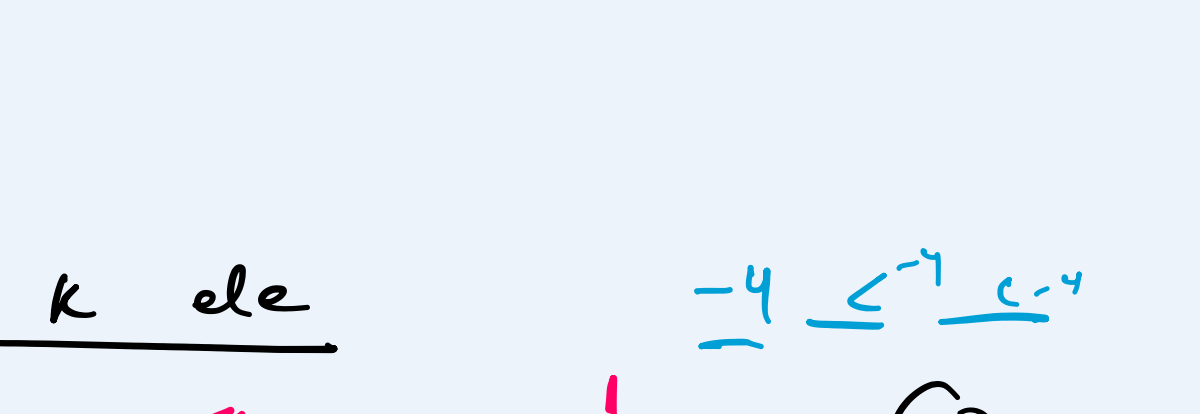
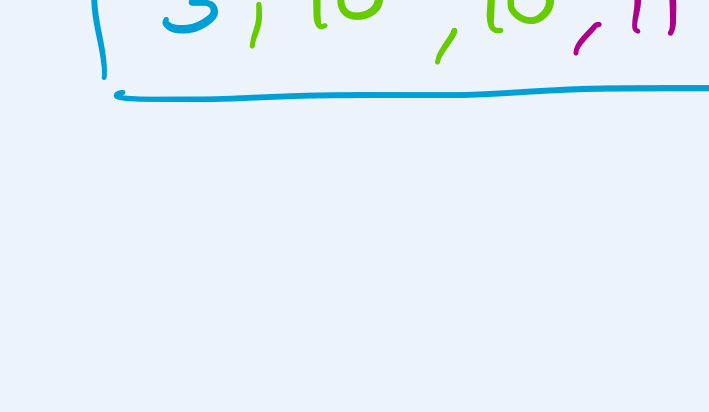
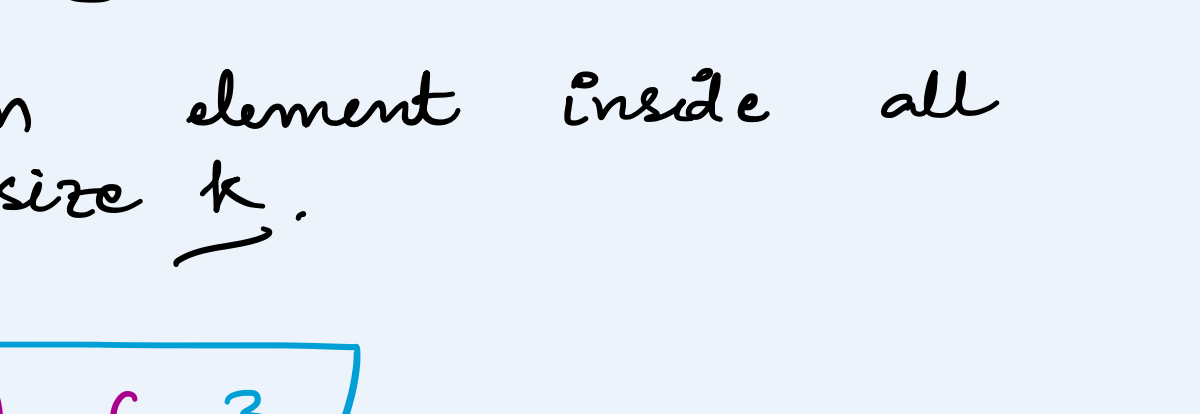
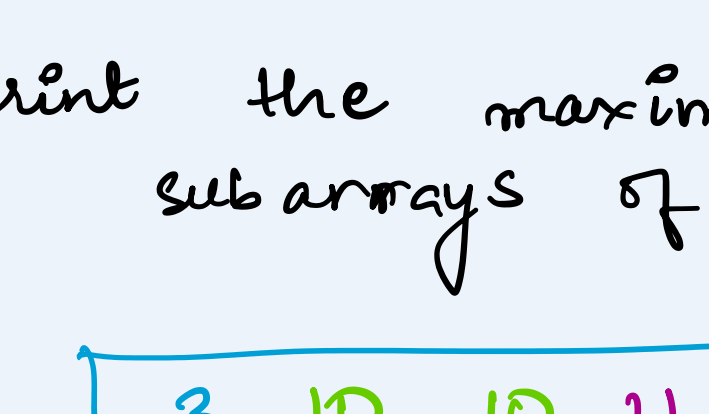
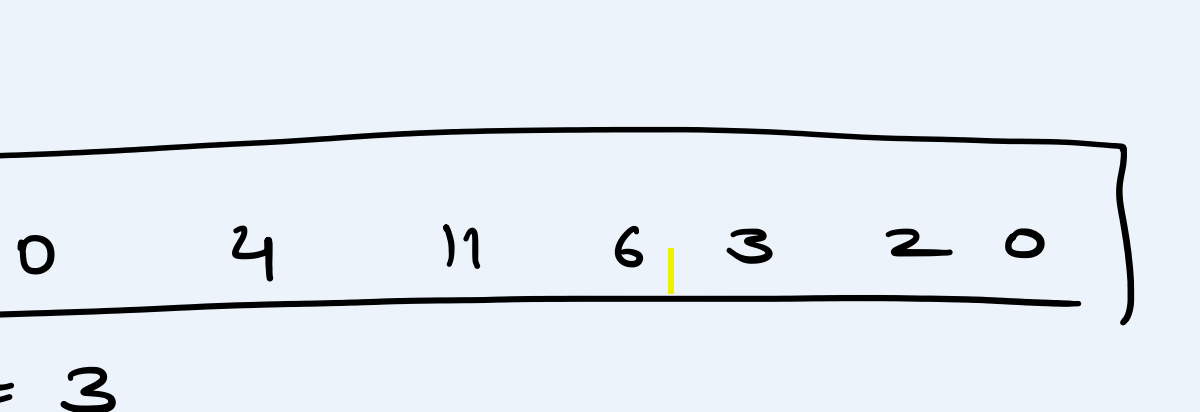
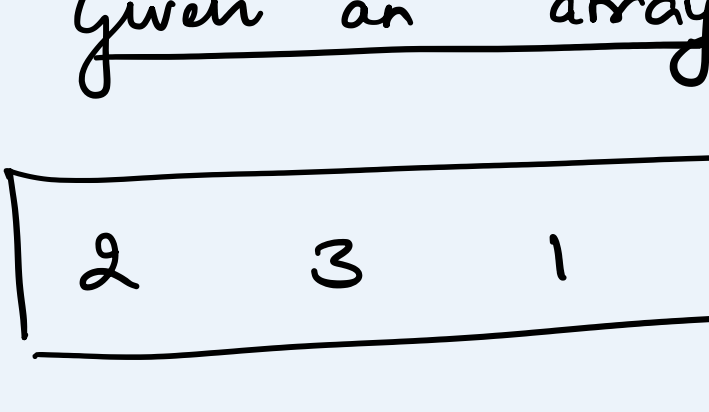
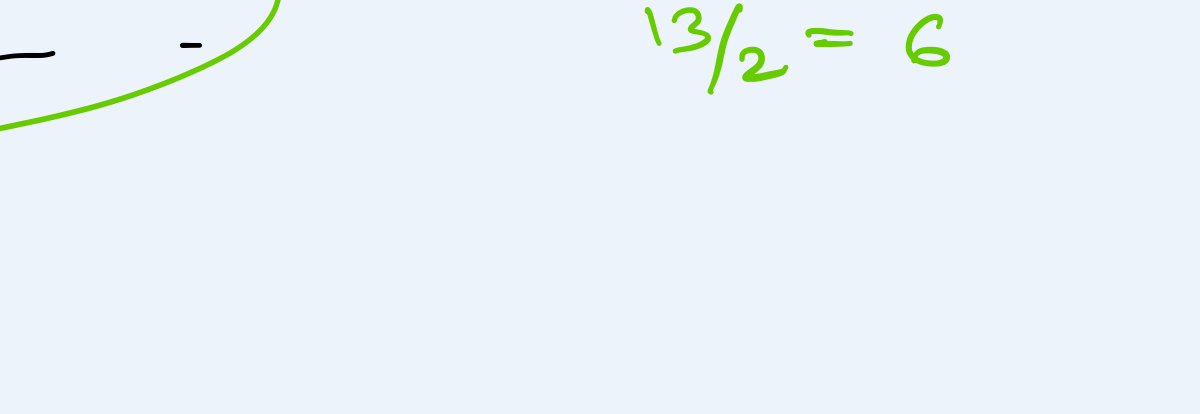
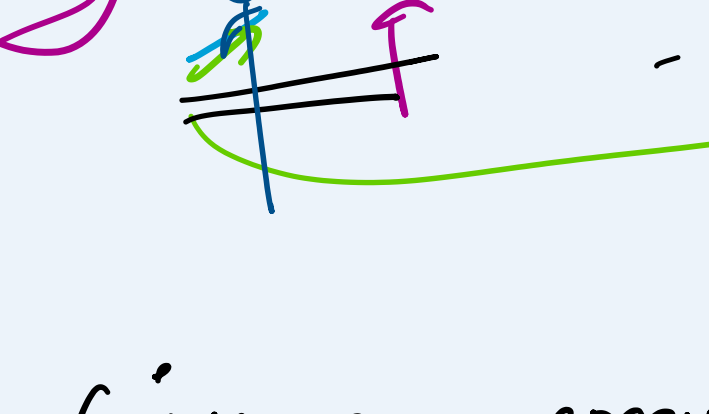
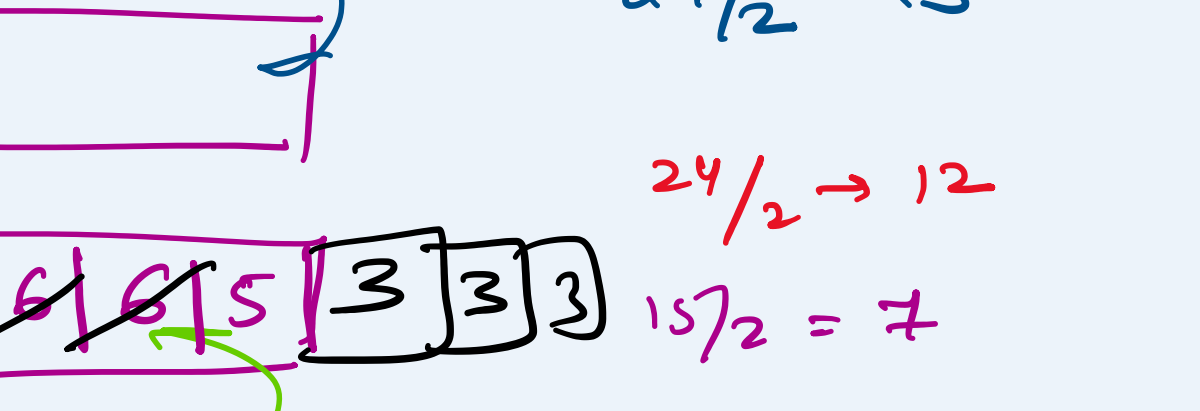
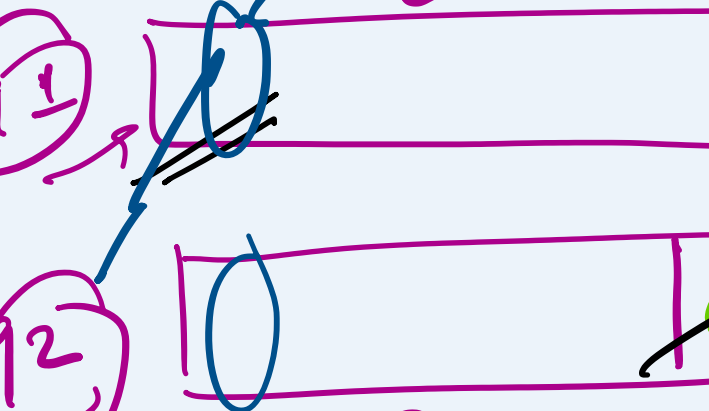
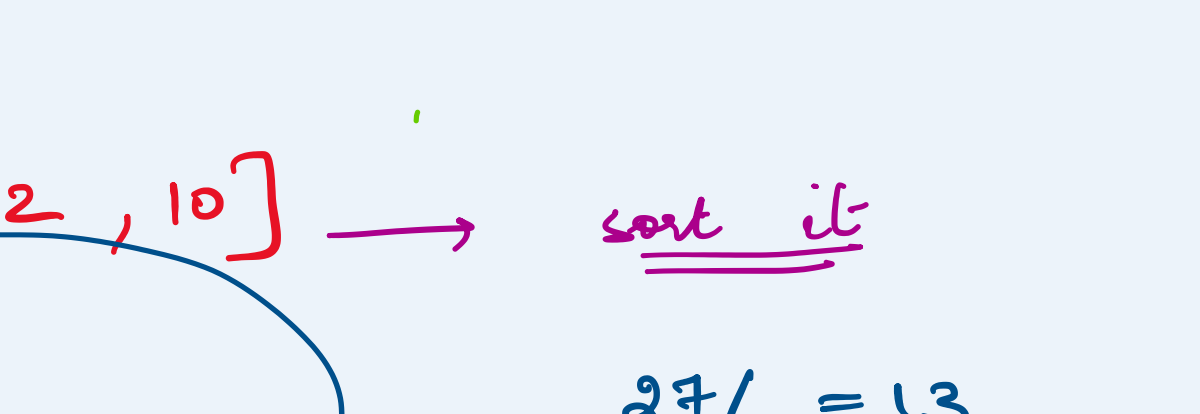
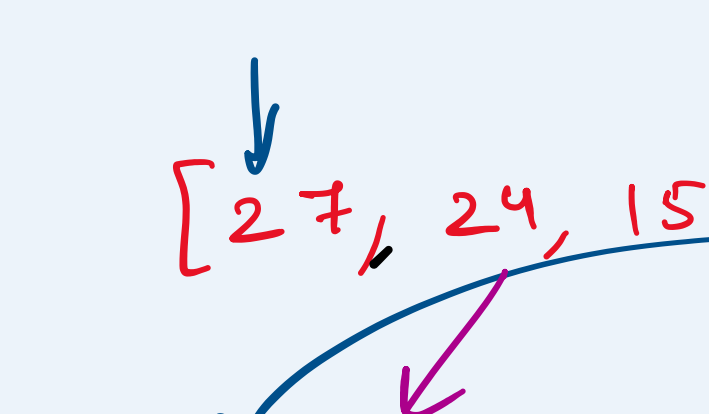
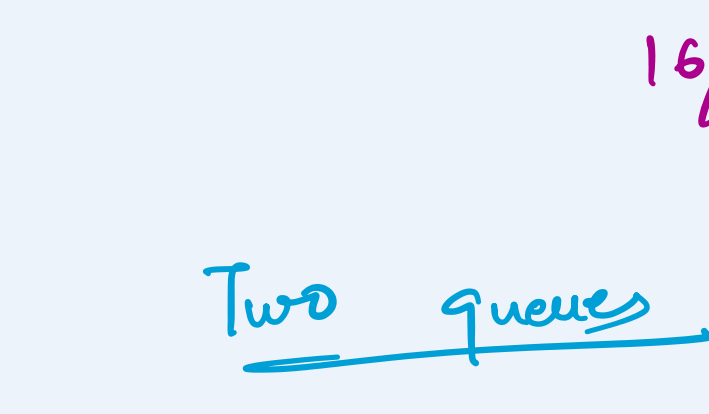
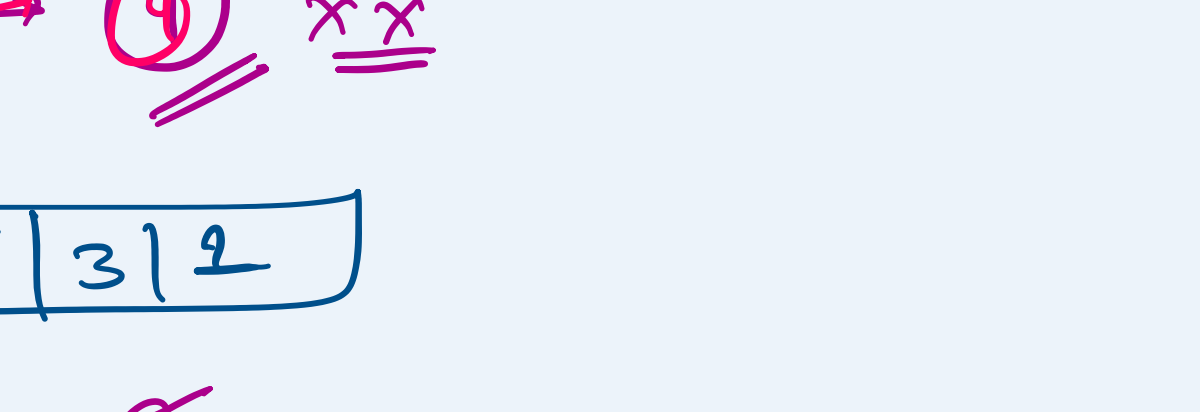
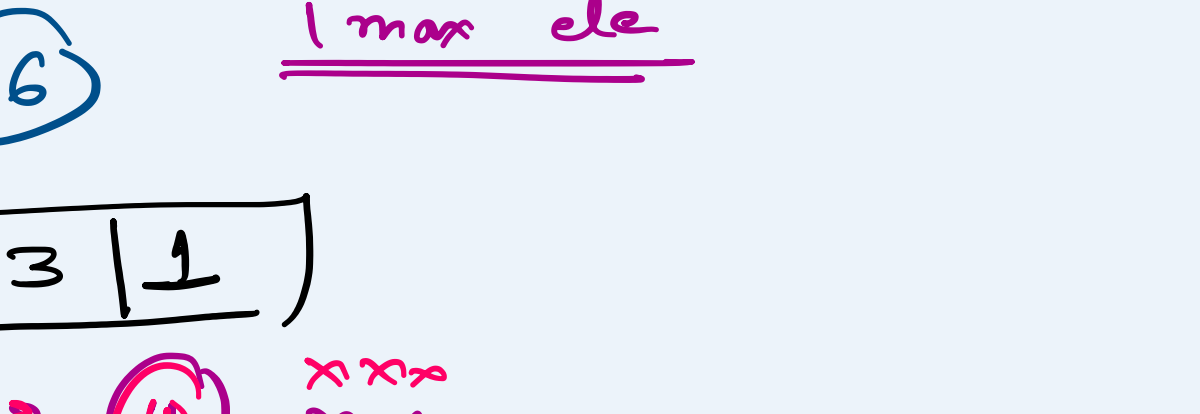
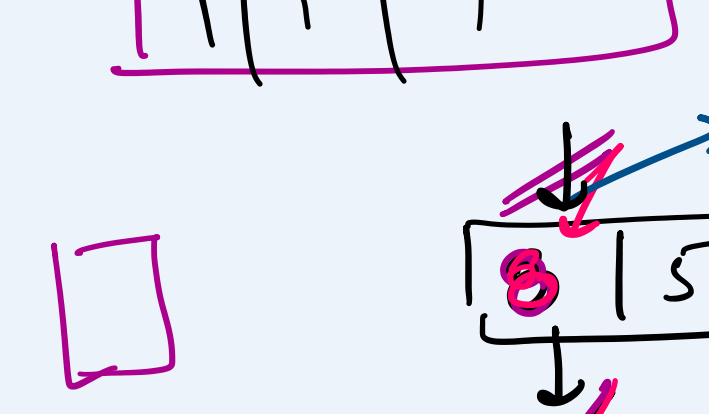
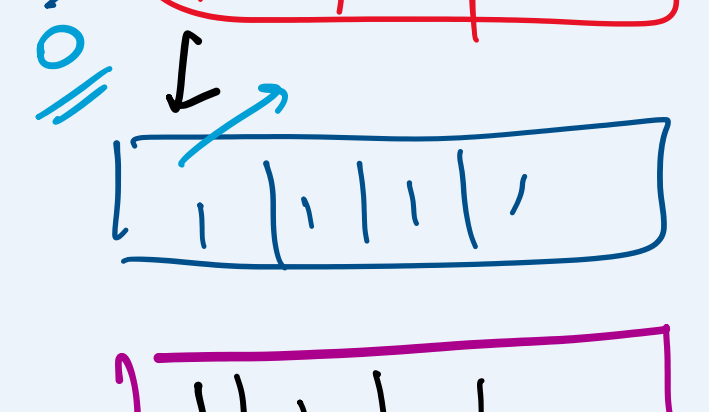
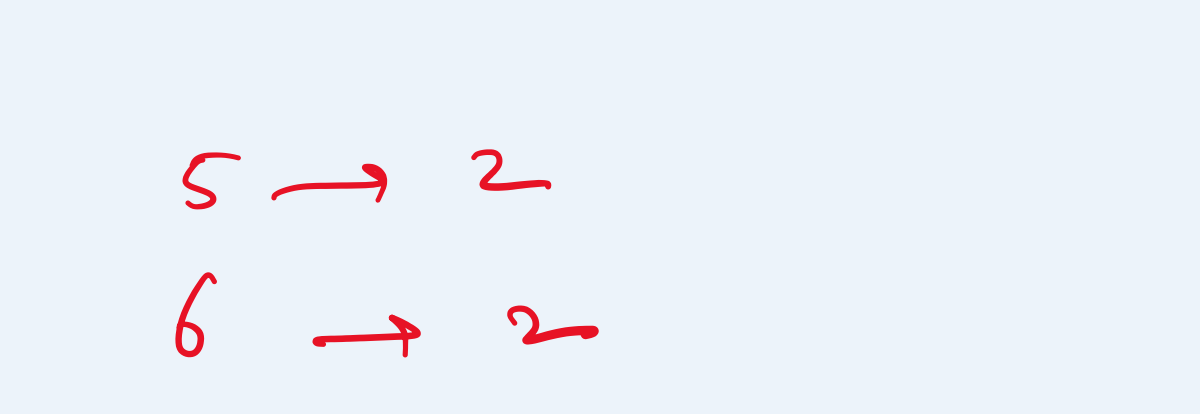
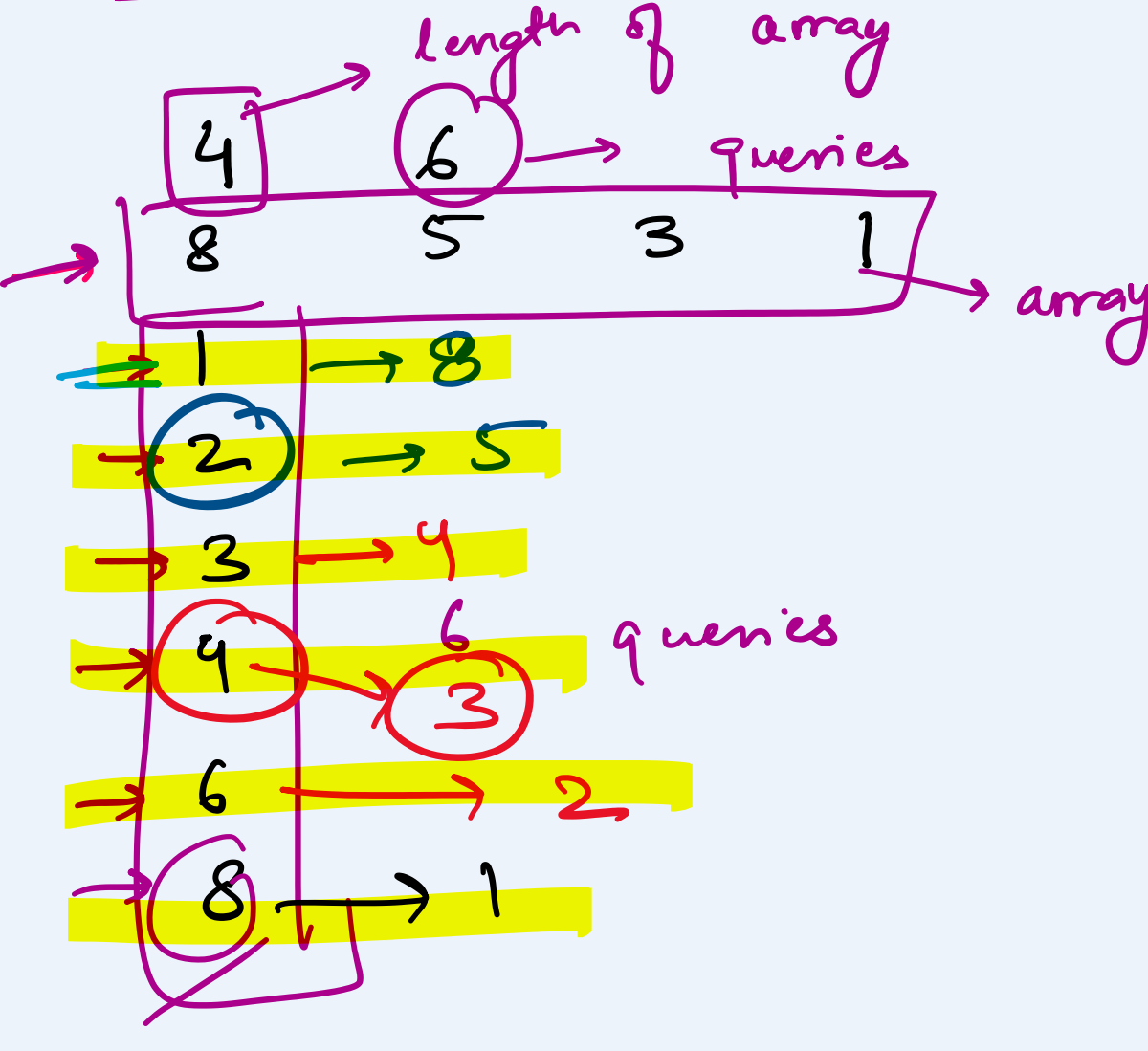
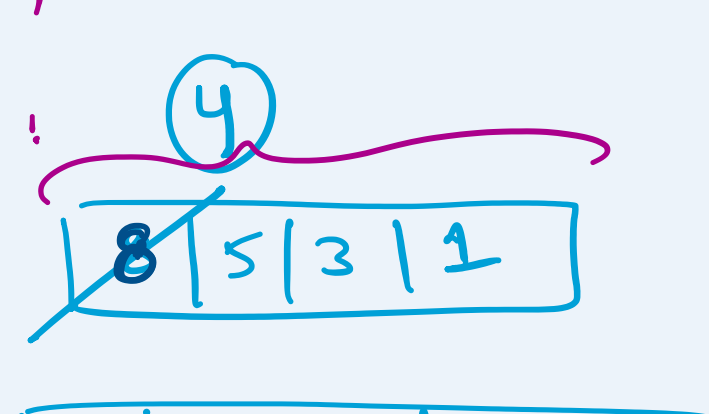
a	1	2
b	1	2
c	1	3

first char of the queue will be the unique char so far.



a → 1
 b → 2
 c → 3
 d → 4

count [str.charAt(i) - 'a']++



Q. Given an array

2 3 1 10 4 11 6 3 2 0

k = 3

Print the maximum element inside all subarrays of size k.

3, 10, 10, 11, 11, 6, 3

Deque

first k ele

-4 ≤ -1 ≤ 4

2, 3, 7, -10, -4, 11, 6, 3, 2, 0

3 2 0

PP

1 2 3 7

PP

3 7 -10

-10, -4,

3 7 -10

7 10 -4

-10 -4 11

3