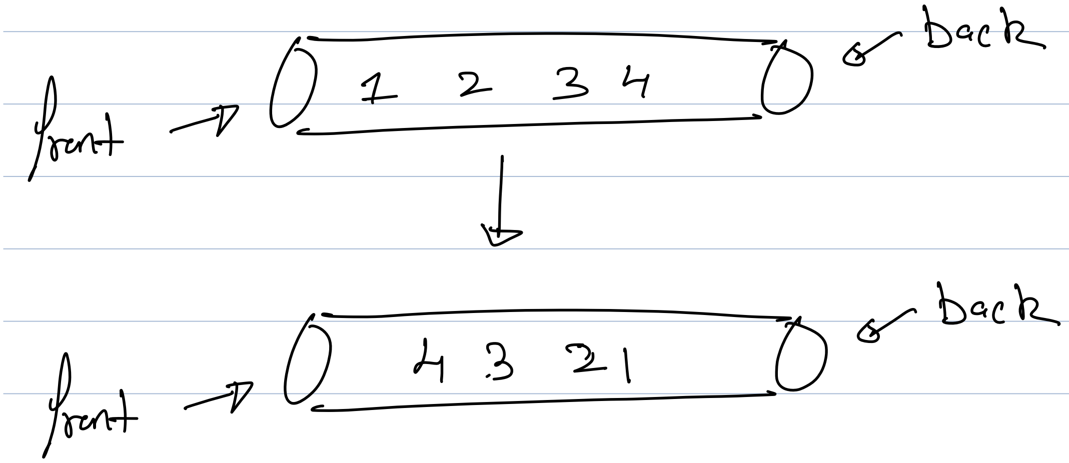
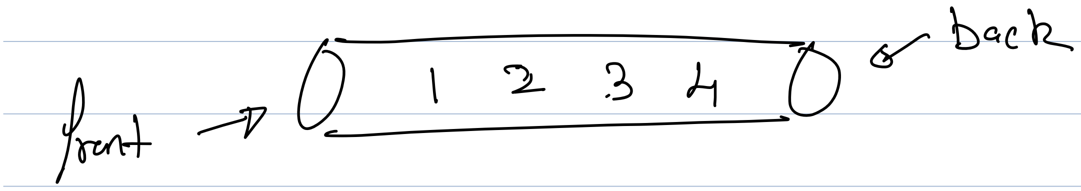
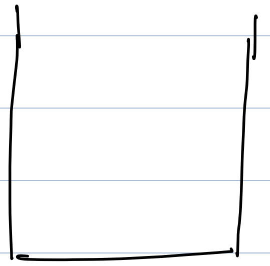
Q1 Given a Queue. Reverse it.



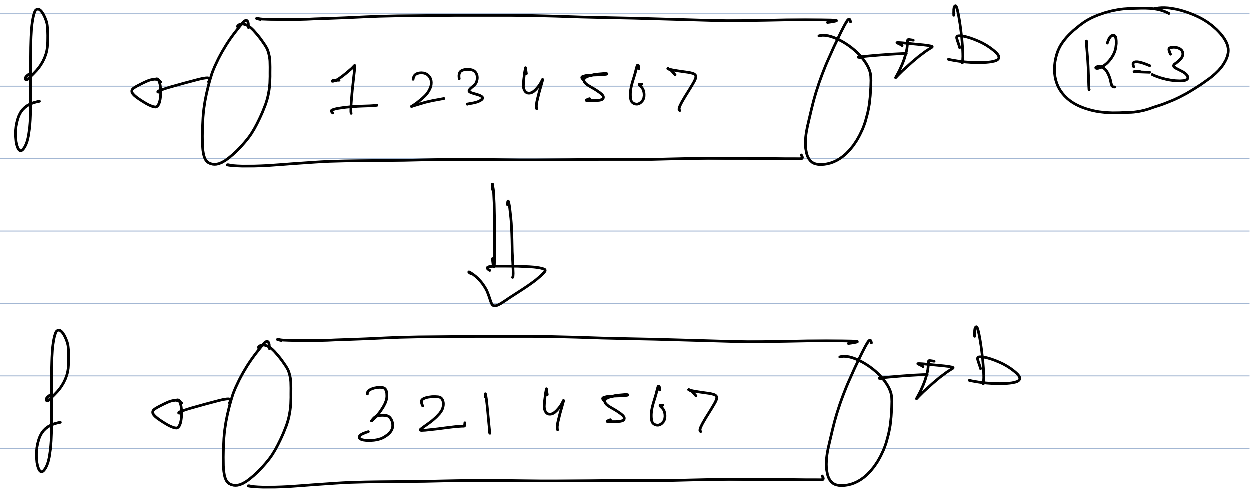


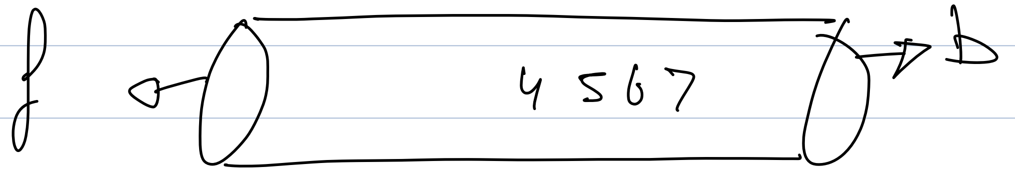


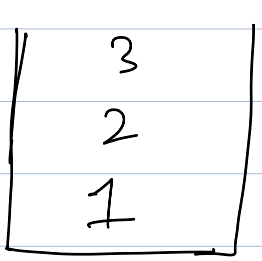
Tc: O(n)
Sc: O(n)

YOU CANNOT TRAVERSE IN A STACK OR QUEUE.

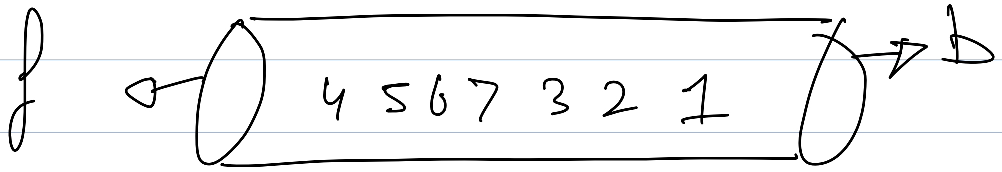
Q2 Given a Queue. Reverse the first K elements.

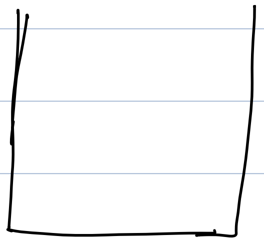






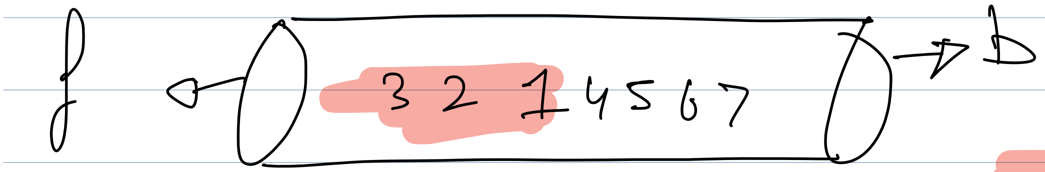
1) Remove first K elements & insert in a stack.





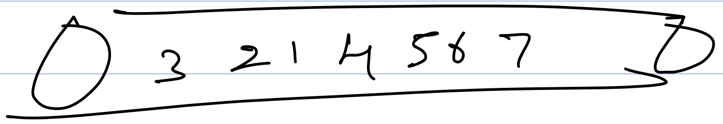
2) Pop from stack,
push to Queue.

3) Deque & Enque in the Queue (n-k) times



TC : O(n)
SC: O(n)

1	2	3	4	5	6	7



Q3 Given a string. Find the first non-repeating character.

⇒ 26 unique character.

Ex1 abcabcef.

Approach 1 : Using ⇒ maps.

1) traverse and maintain of each character.

2) traverse the string again & find the first character where freq is 1.

TC : n+n ⇒ $O(n)$

SC : $O(K)$

Tc : n+k   Sc : O(k)

0	1	2	3	4	5	6	7	8
e	c	h	e	d	f	c	a	c

a	b	c	d	e	f	g	h	i
7	-1	-2	4	-2	5	-1	2	-1

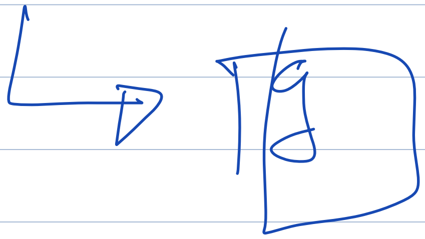
-1 ⇒ the character hasn't been encountered.

-2 ⇒ character has occured more than once.

Tc : O(n+k)

0	1	2	3	4	5	6	7	8
a	c	a	c	f	g	f	h	i

a	b	c	d	e	f	g	h	i	j	k
-2	-1	-2	-1	-1	-2	5	7	8	-1	-1



Q4 Given a stream of characters, after adding every character. find the first non-repeating element.

Ex1

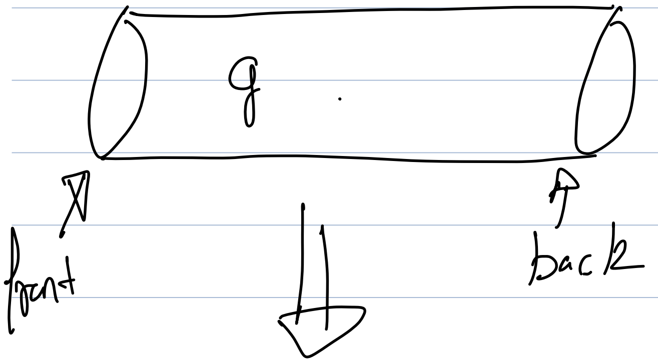
a	b	c	a	d	e	d	a	b	e	c	g
a	a	a	b	b	b	b	b	c	c	#	g.

Approach 1 : Using the last Solution after every new character.

TC : $O(nk)$ SC: $O(k)$.

a b c a d e d a b e c g

a a a b b b b b c c # g.

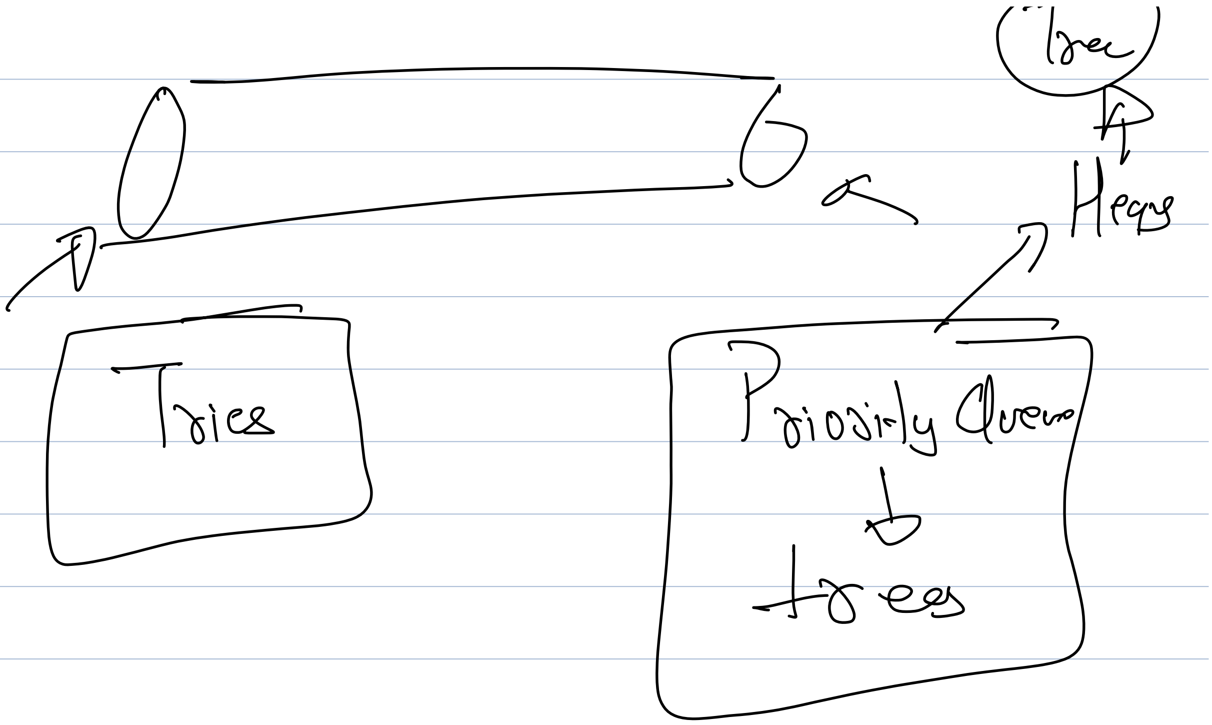


Candidates which can be ~~or~~ answers

a → 3
b → 2
c → 2
d → 2
e → 2
g → 1

Tc: O(n+k)

Sc: O(K)


Tree
Heaps
Tries
Priority Queue
trees