

---

---

---

---

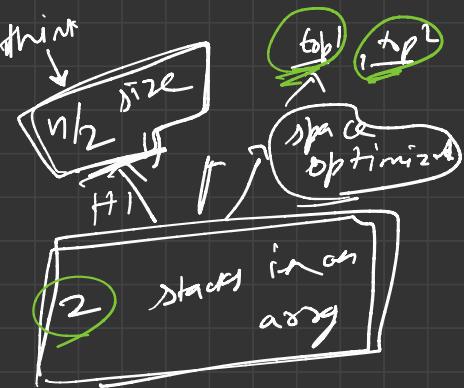
---



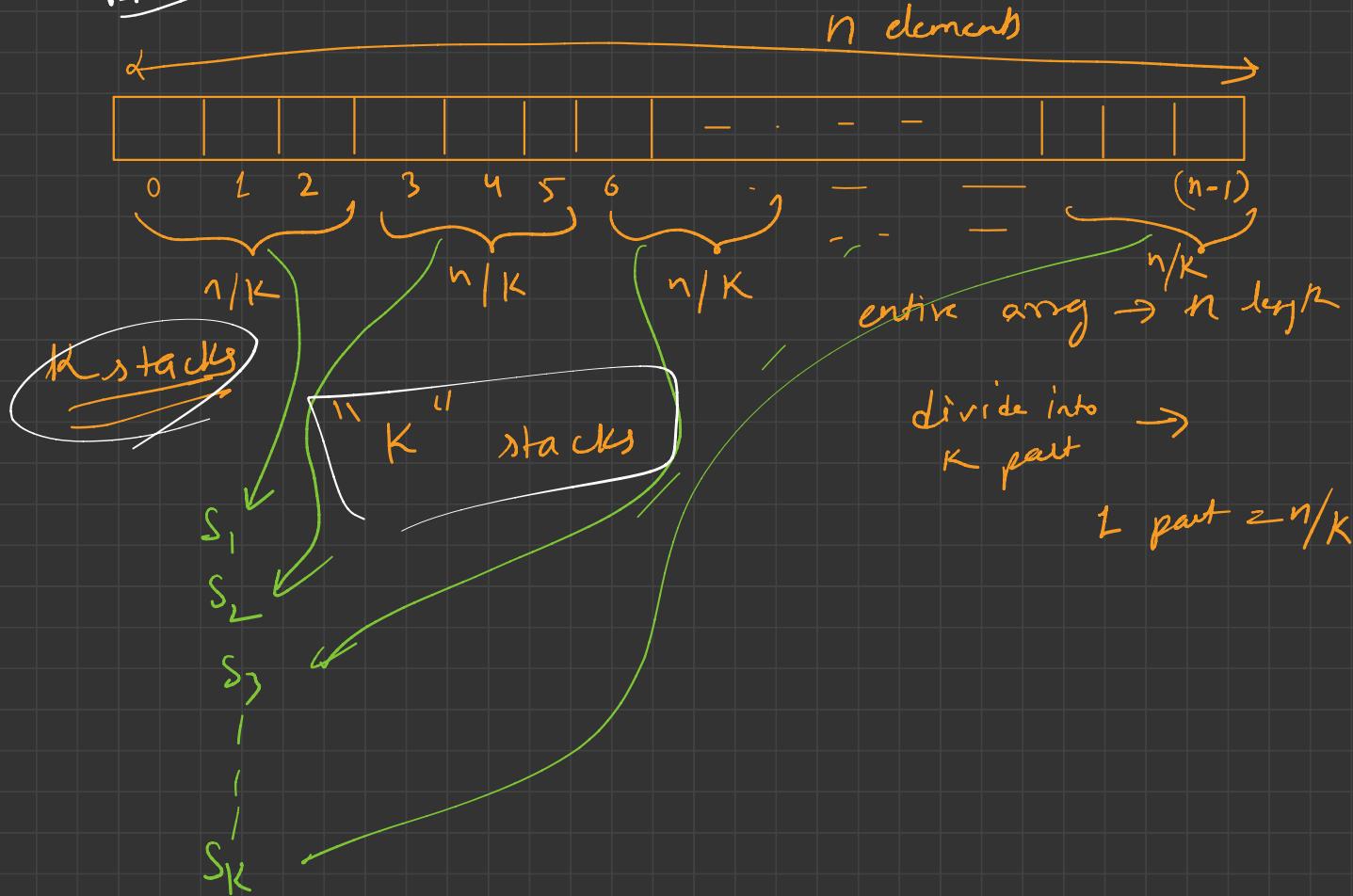
# Stacks

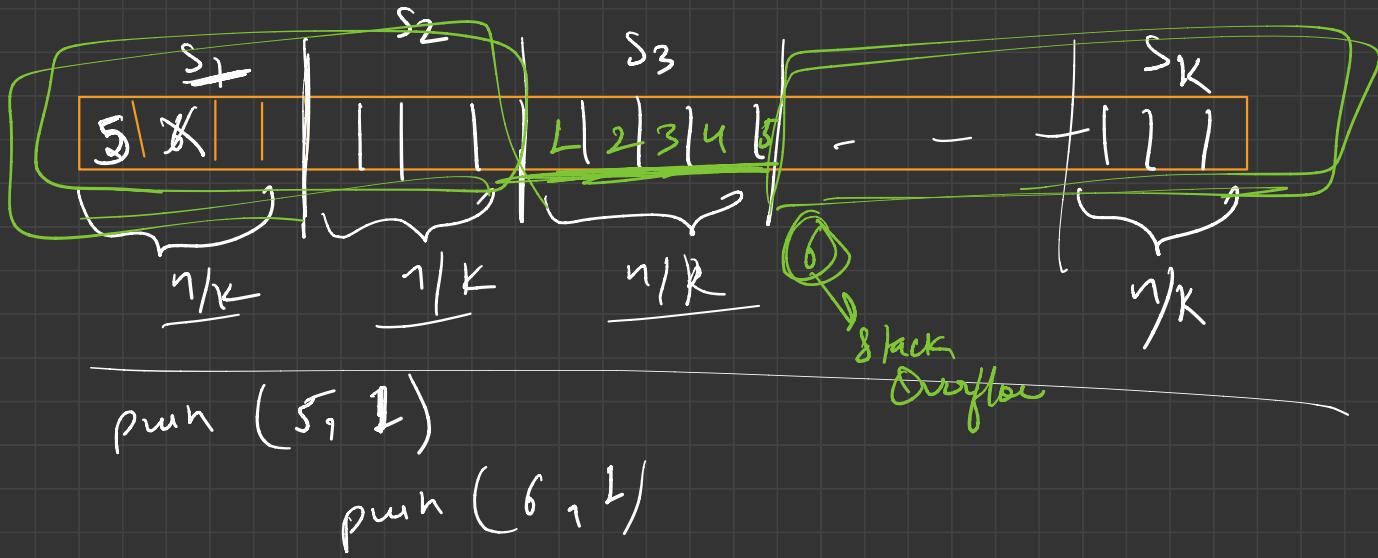
HARD

$\rightarrow$   $N$  stacks in an array



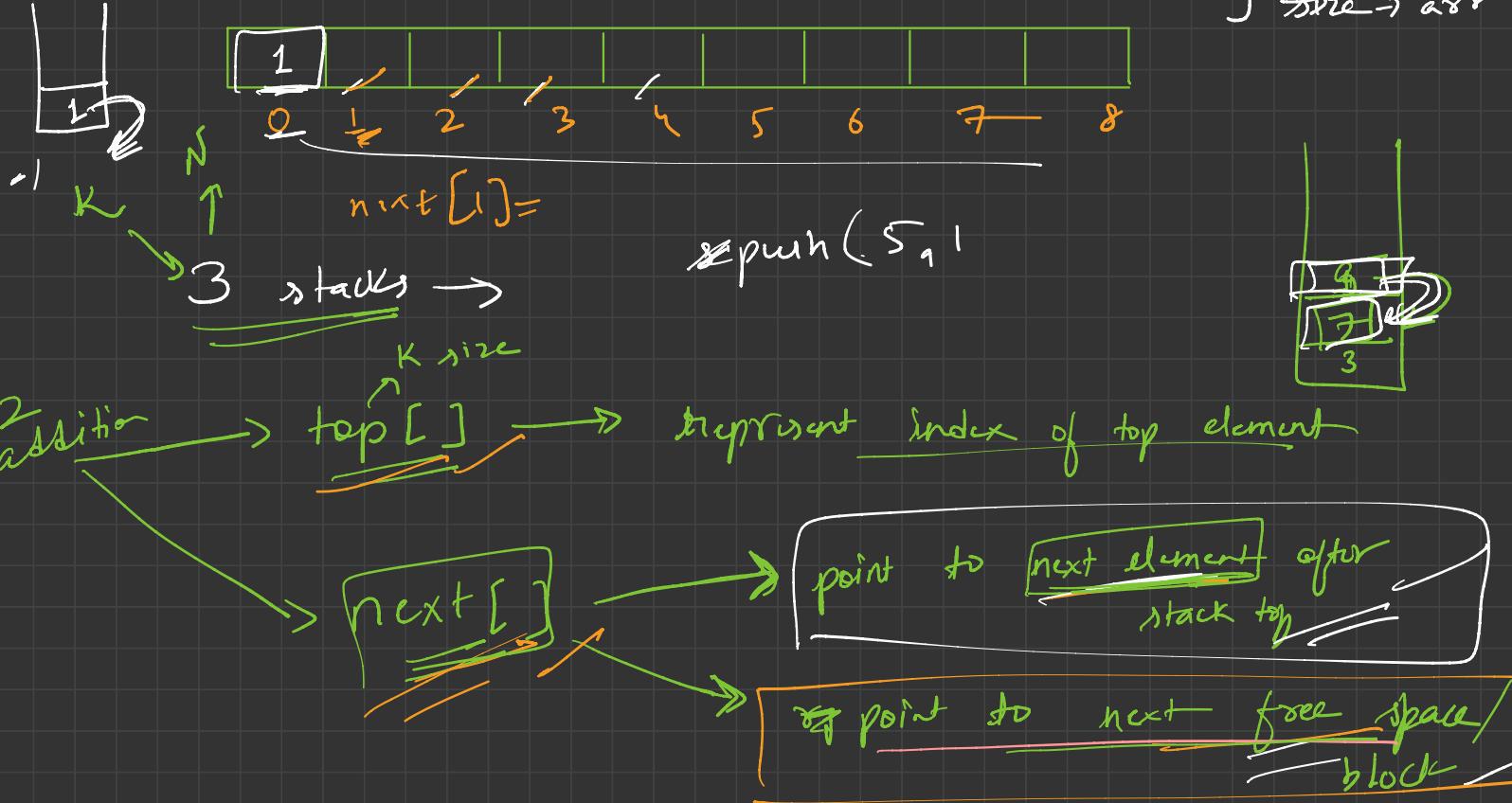
→ Approach :-





Drawbacks: - Space optimised  $\rightarrow$  ~~X~~

## Approach #2



$i \in (\text{arr}[:]) \rightarrow$  store an element

$\text{arr} \rightarrow \text{arr}[:] \rightarrow \text{nothing stored}$

$N \rightarrow$  no of stacks  $\rightarrow 3$

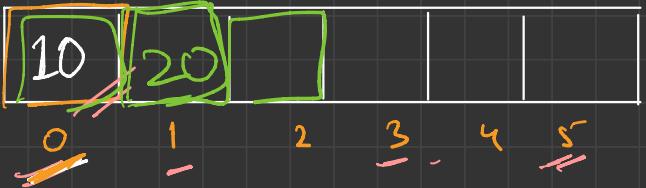
$S \rightarrow$  size of array  $\rightarrow 6$

$Q \rightarrow$  no of queries  $\rightarrow 5$

$s_1, s_2, s_3$

array

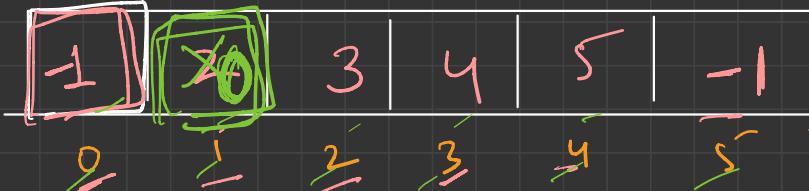
arr →



top →



next =



freeSpot =



push(10, 1)  
push(20, 1)

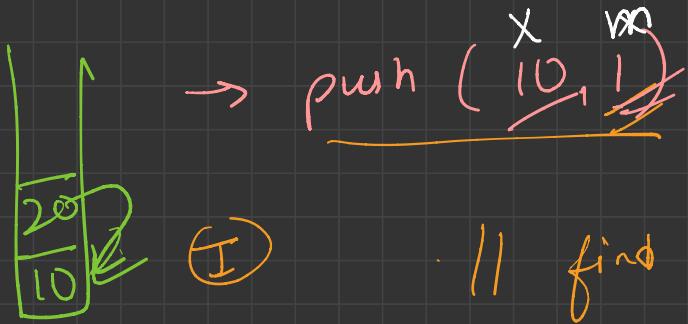
push(30, 2)

pop(1)

pop(2)

constructor





(I)

// find index =

int index = freeSpot;

(II)

// freeSpot update

freeSpot = next [index];

(3)

// insert in array

arr [index] = x;

(I) index = 0

(II) freeSpot = next [0]

freeSpot = 1

(III) arr [0] = 10

(IV)

next [0] = top [0]

(V)

top [0] = 0



(4)      // update next      pushon top

next [index] = top [m - 1];

(5)      // update top

top [m - 1] = index;

push (20, 1)

index = 1  
front = next [index] = next [1]  
front = 2

↳  $\text{arr}[1] = 20$

↳  $\underline{\text{next}[1]} = \underline{\text{top}[0]}$

↳  $\text{top}[0] = 1$

T.C  $\rightarrow O(1)$ ,  $s + n + s$

S.C  $\rightarrow O(s + n)$







