

Stack

09 October 2022 13:06

LIFO

Last in first out

C++

push

pop

top

empty

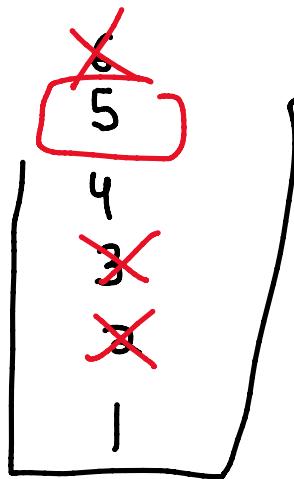
Java

push

pop

peek

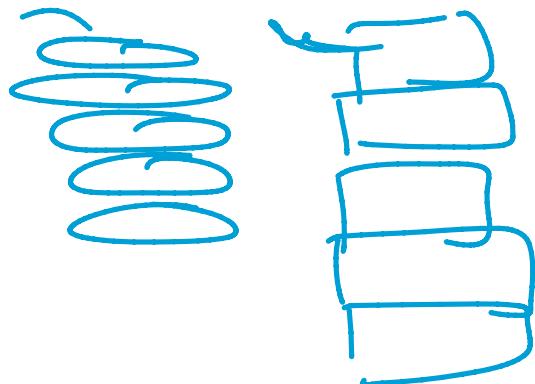
isEmpty



3, 2, 6,

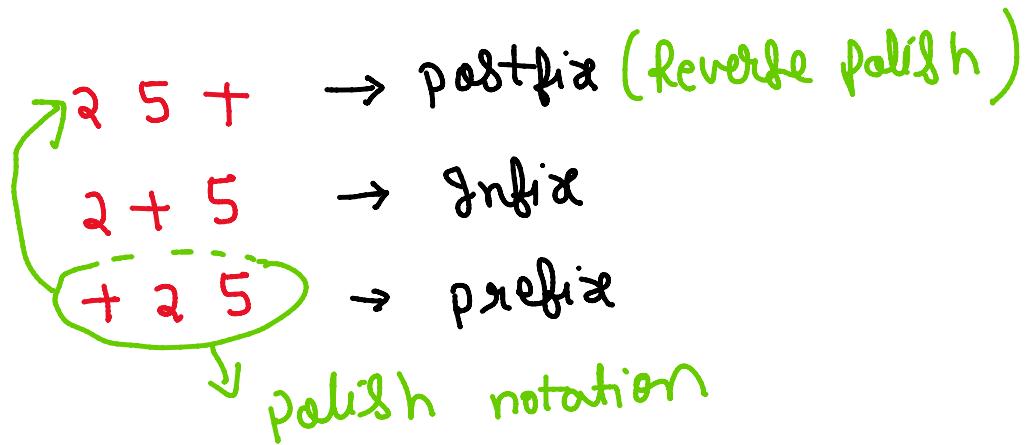
stl, collections

undo



Reverse polish notation

09 October 2022 13:14



$2\ 5\ +$

$\textcircled{+}$

$\checkmark\ \checkmark\ \checkmark$
 $2\ 5\ 3\ +\ +$

$2\ 8\ +$
10

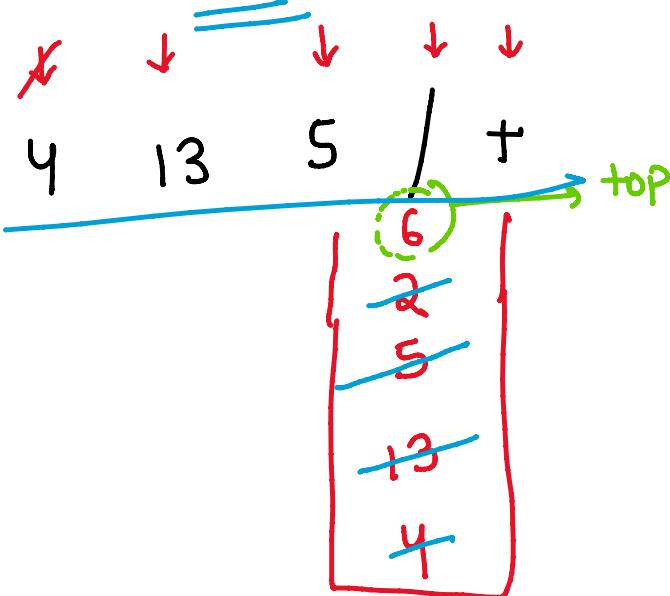
$2\ 1\ +\ 5\ *$

$2\ 12\ 4\ /+$

$3\ 5\ *$

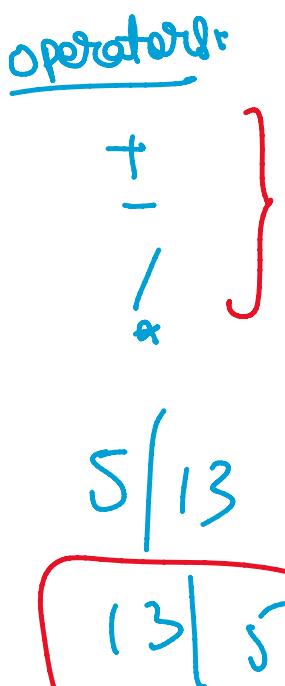
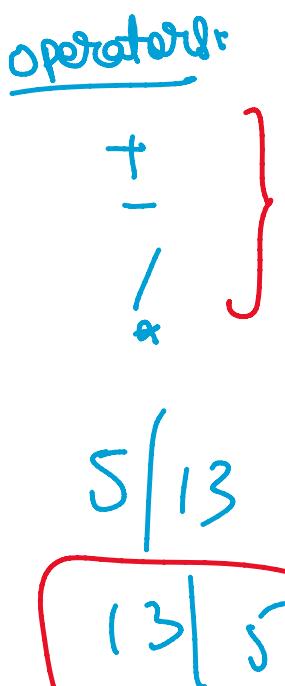
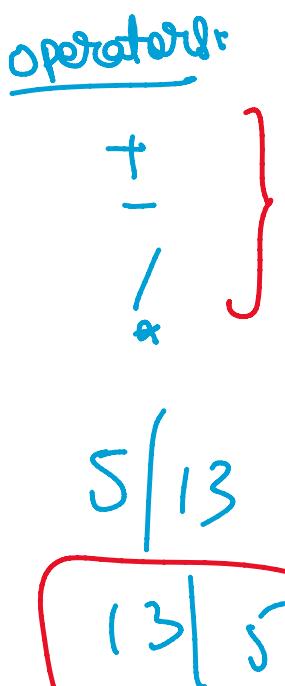
$2\ 3\ +$
 $\underline{\underline{5}}$

$\underline{\underline{15}}$



$$\begin{aligned} \text{int } x &= 5 \\ \text{int } y &= 13 \end{aligned}$$

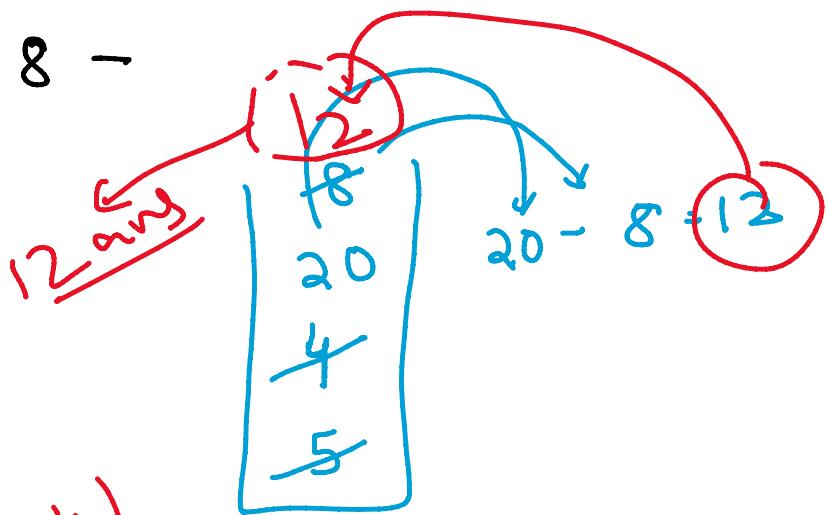
$$13/5 = 2$$



$\begin{matrix} & \downarrow & \downarrow & \downarrow & \downarrow \\ 5 & 4 & \times & 8 & - \end{matrix}$

$$\begin{matrix} & \downarrow & \downarrow & \downarrow \\ y & + & z & = 6 \end{matrix}$$

$$\begin{array}{|c|c|} \hline 13 & 5 \\ \hline y & z \\ \hline \end{array}$$



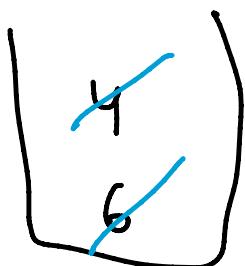
$O(N)$
 $O(\text{no. of operators})$

$$5 * 4 = 20$$

$\begin{matrix} & \downarrow \\ 6 & 4 & - \end{matrix}$

$$x - y$$

$$\begin{array}{|c|} \hline y - x \\ \hline \end{array}$$



$$\begin{matrix} x = 4 \\ y = 6 \end{matrix}$$

Valid Parenthesis

09 October 2022 13:39

$() [\{ \}] \}$ → invalid

$[()] \} \{$ → invalid

$([\{ \}]) \{ [] \}$ → valid

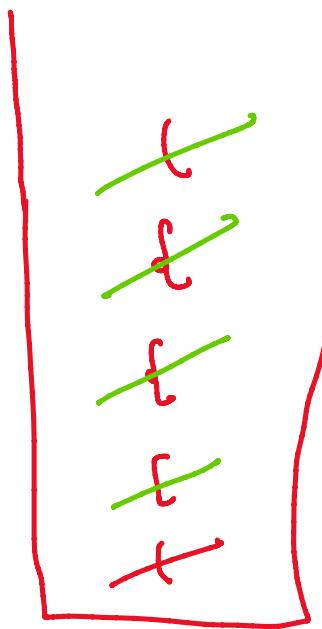
1. push the open brackets.

2. and for closing bracket, check stack top should

same bracket of opening.

$([\{ \}]) \{ () \}$ → true

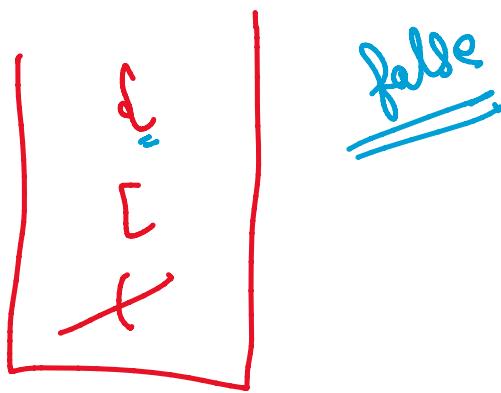
stack <char>



$() [\{ \}] \}$ → false

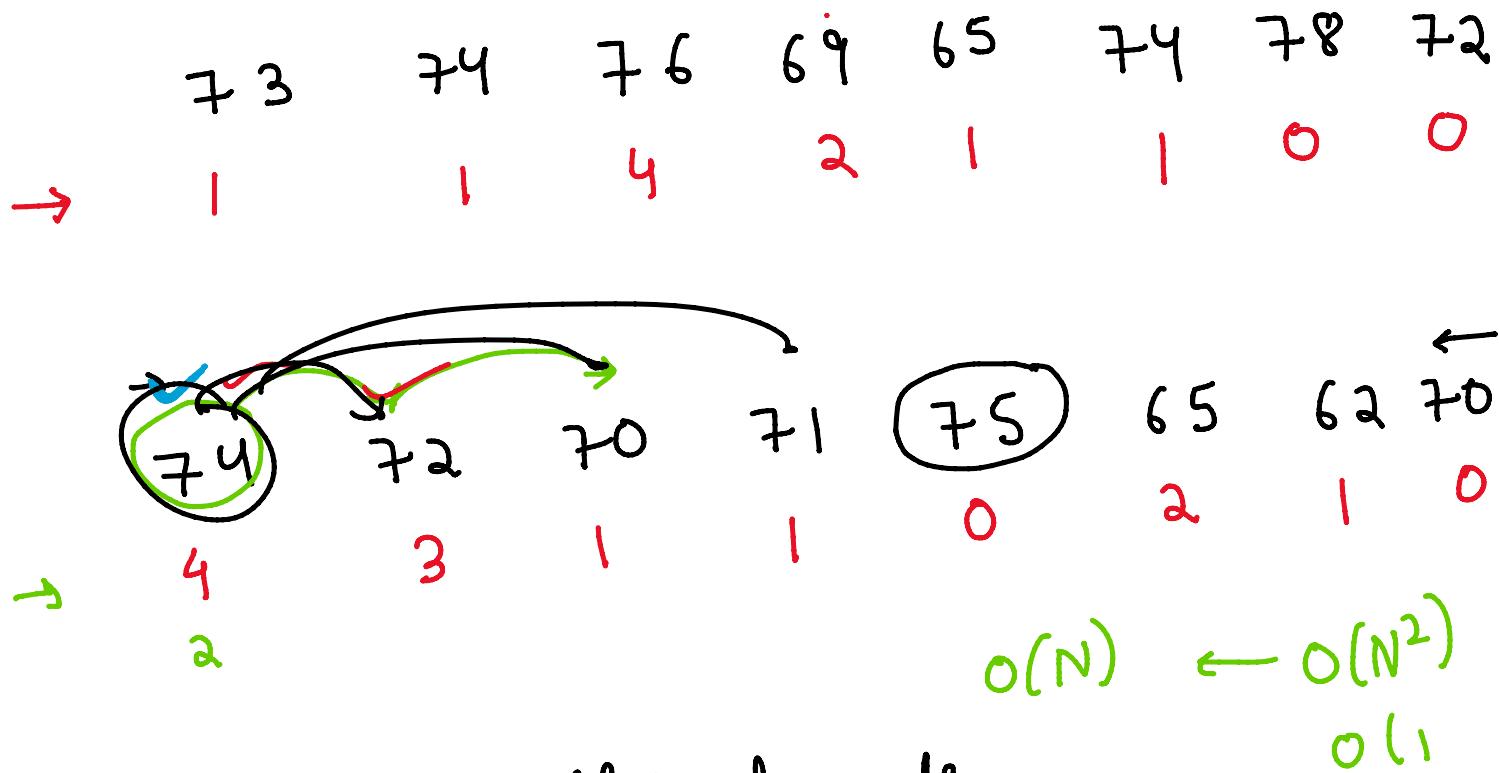
() [{ }] \rightarrow ~~fun~~

$O(N)$
 $O(\text{no. of m})$



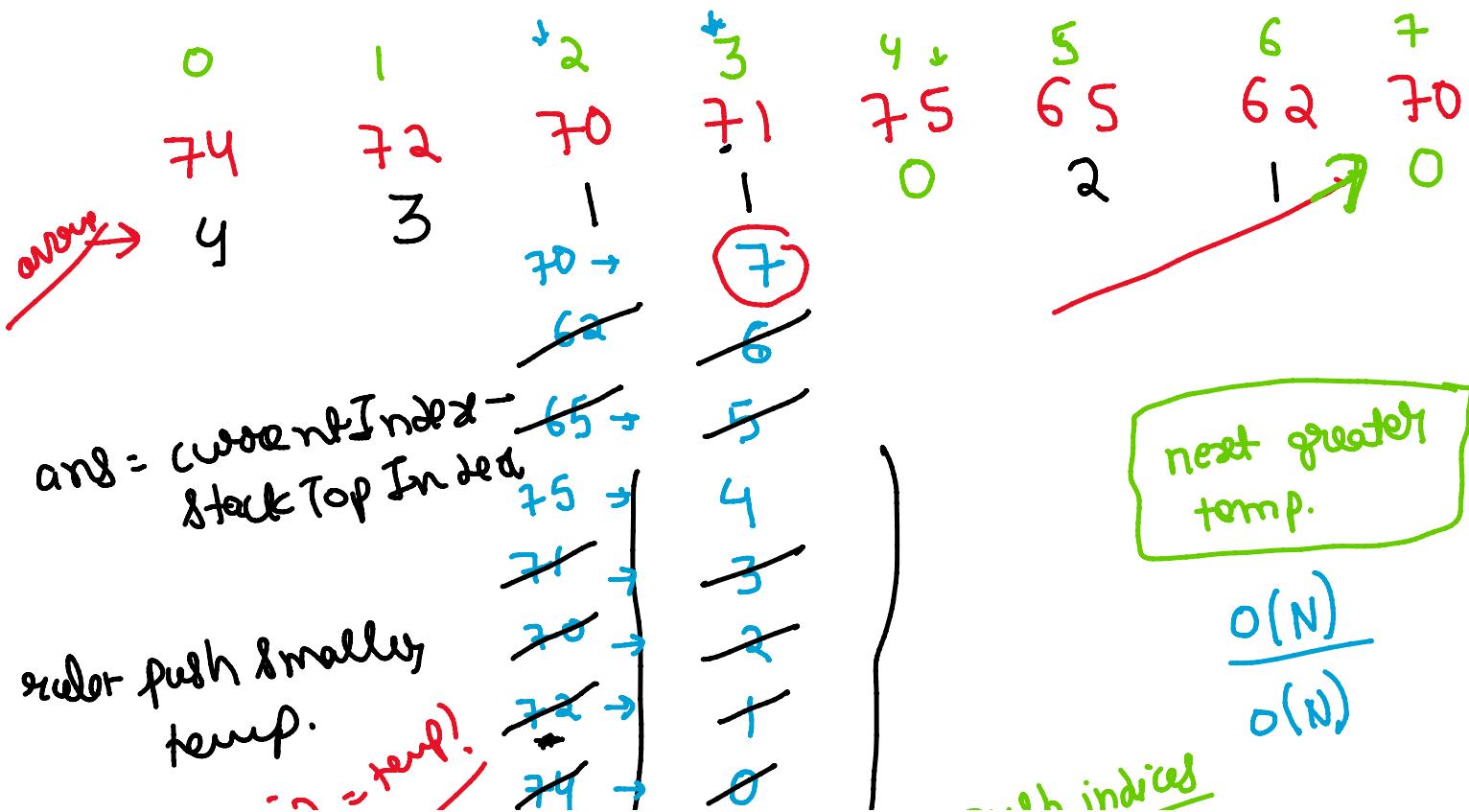
Daily temperatures

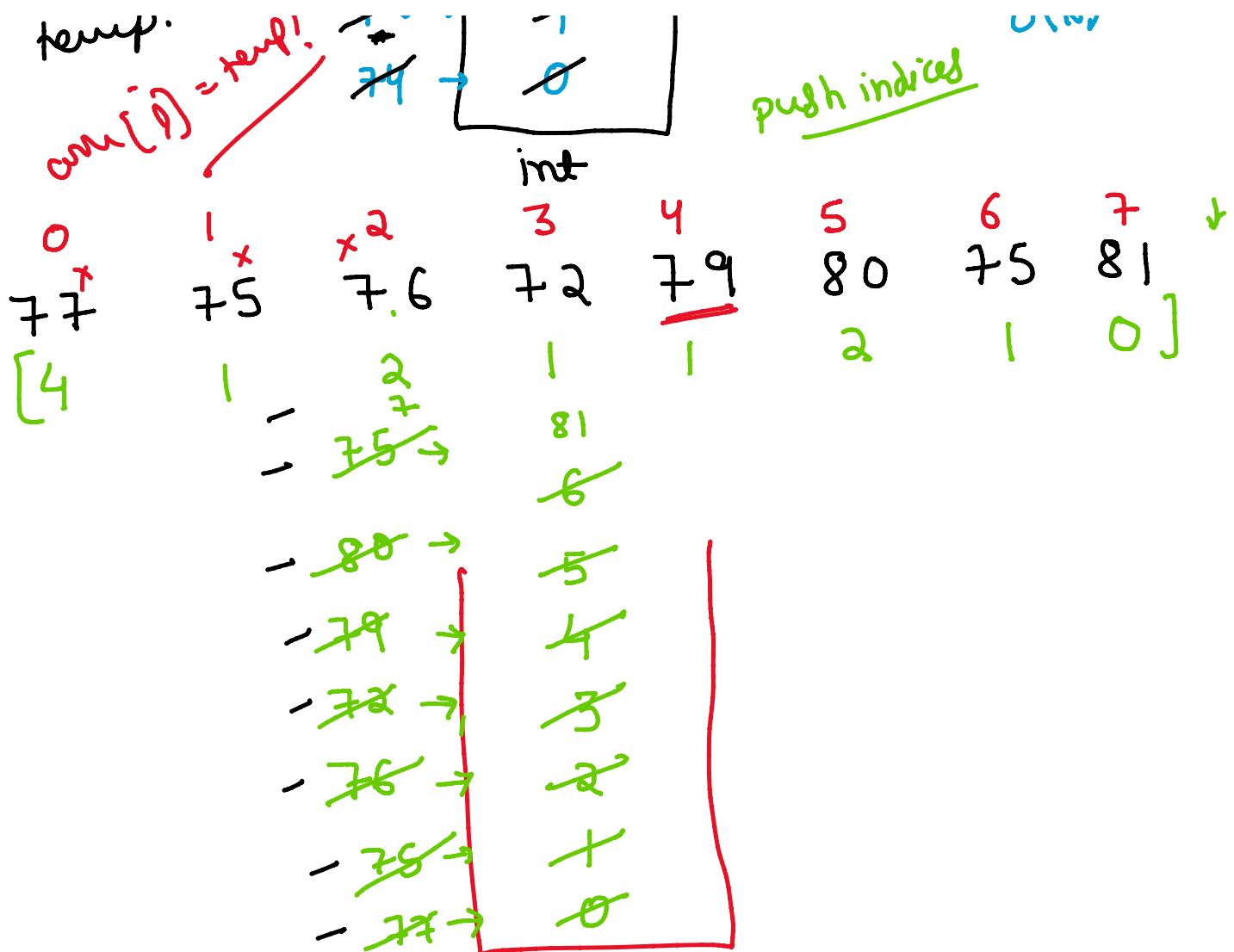
09 October 2022 14:00



1. push? = push smaller elements.

2. pop! :





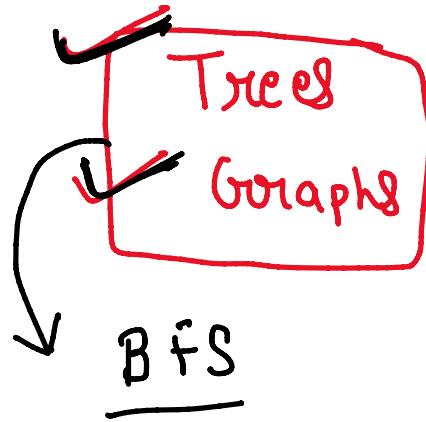
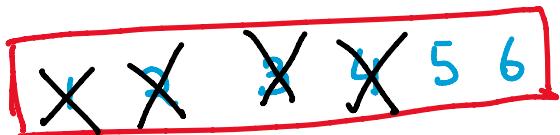
- 1. Next greater element in an array.
- 2. Next smaller element
- 3. Stock span problem

Queues

09 October 2022 15:01

FIFO

first in first out



1, 2, 3, 4

C++

push
pop
front
empty

Java

add
remove/poll
peek
isEmpty



Implement Queue using Stacks

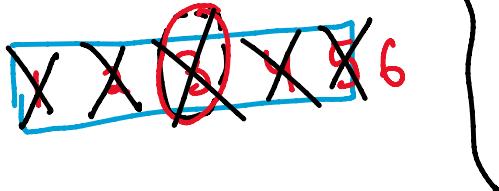
09 October 2022 15:08

implement



ans = 1

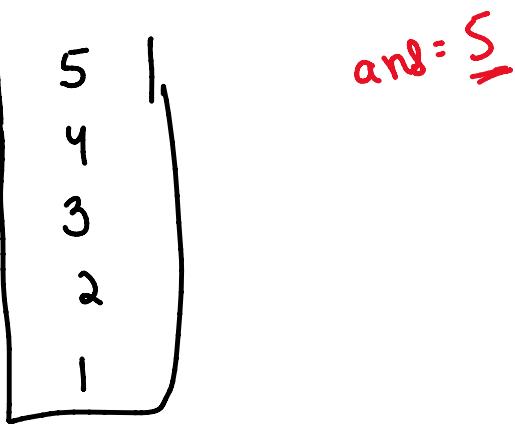
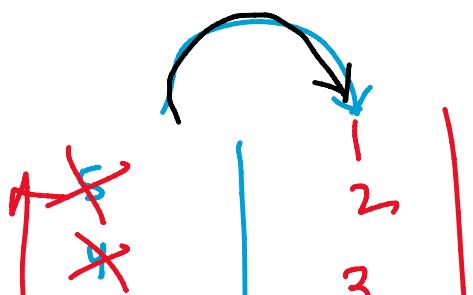
ans = 1, 2, 3, 4



1. push in S1

2. Pop from S2.

(if S2 is empty,
take all values from
S1 to S2).

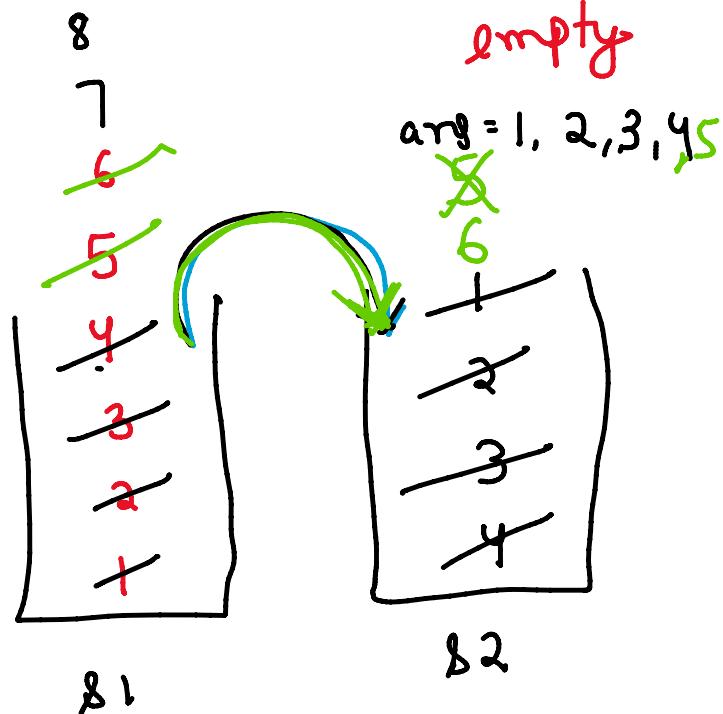


push

pop

peek

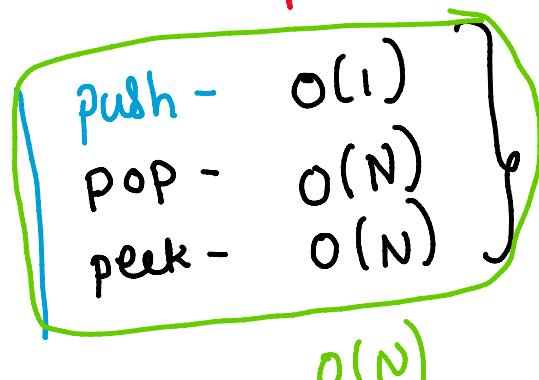
empty



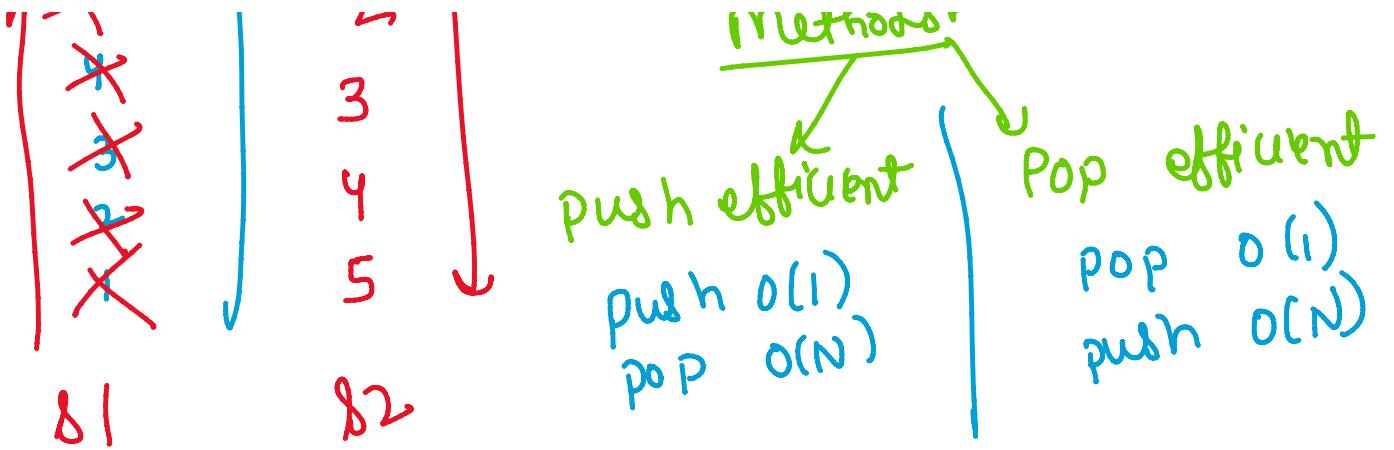
S2

(push)

(pop
peek)

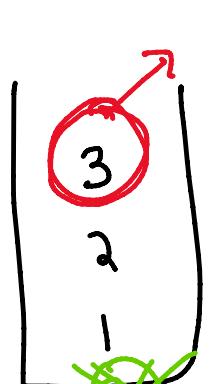


Methods

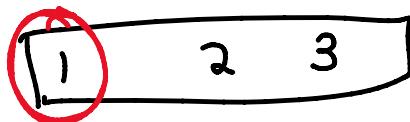


Implement Stack Using Queue

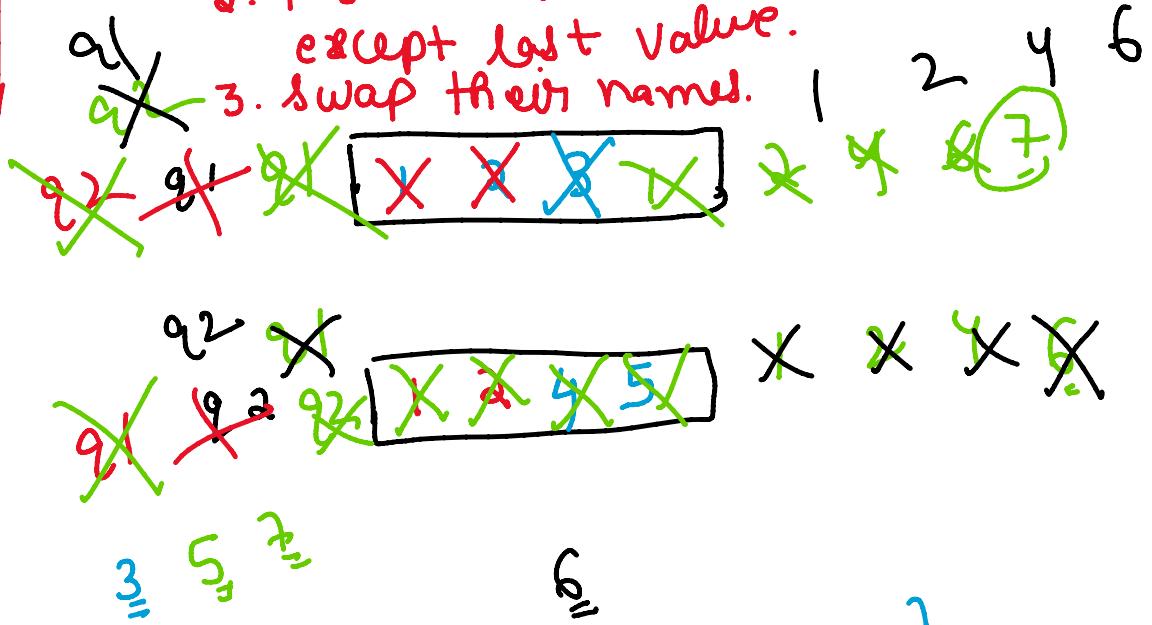
09 October 2022 15:25



using



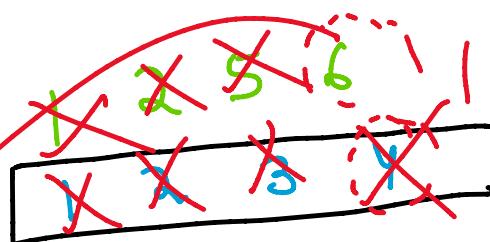
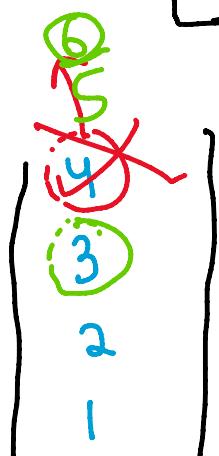
1. push always in q_1
2. push all values from q_1 to q_2 except last value.
3. swap their names.



$\text{push} - O(1)$
 $\text{pop} - O(N)$
 $\text{top} - O(N)$

.size()

single queue?

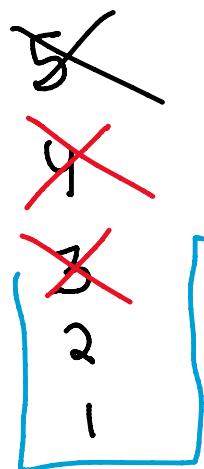


$\dots \text{size}$
 $\text{if } \text{while } (\text{size} - 1)$
 pop



while ~
pop
push
y size

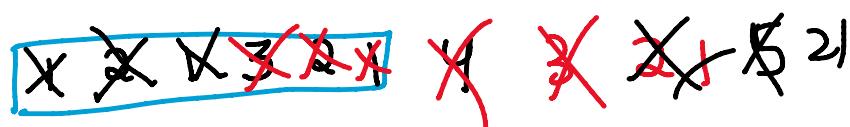
~~4, 3~~ for



- ✓ push - $O(1)$
- ✓ pop - $O(N)$

$$O(N) + O(N) = O(N)$$

$$O(N) \cdot O(N) = O(N)$$

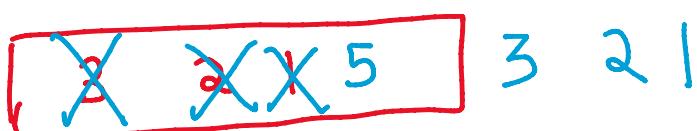


(S)

~~push $O(N)$~~
~~pop $O(1)$~~



5



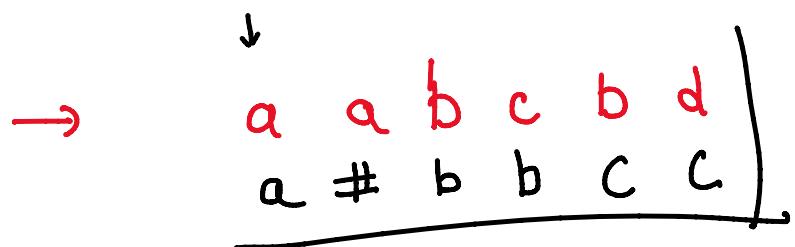
while(size),

size = 3

whole(size)
push }
pop }
==

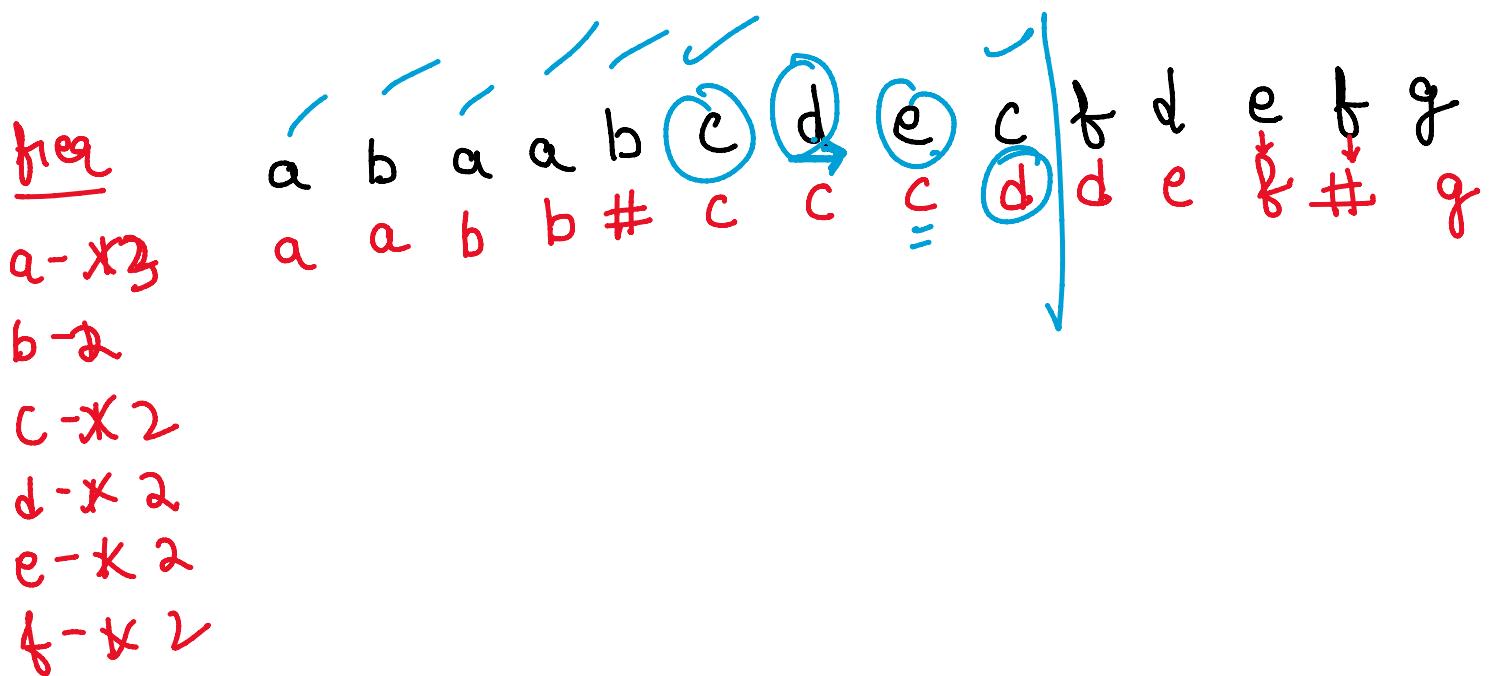
First non repeating character

09 October 2022 15:48



a-2
b-*2
c-1
d-1

queue↑



a a b c b c d d e f
a # b b c c # d # e e

O(N)
O(1)

queue↑

26
a-X2
b-X2
c-*2
d-*2
e-1
f-1

