

Plan

03 September 2023 13:06

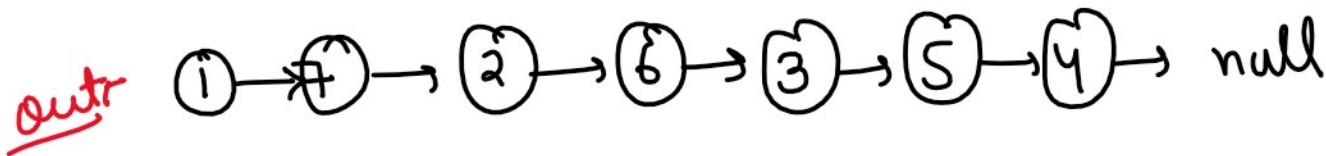
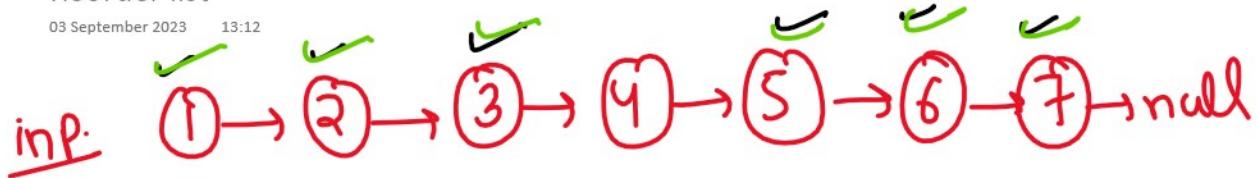


- 1 → Arrays
- 2 → Arrays, Matrix
- 3 → Binary Search
- 4 → Sorting, Strings
- 5 → Recursion
- 6 → Backtracking, Set
- 7 → Hashmap, TC & SC
- 8 → LinkedList

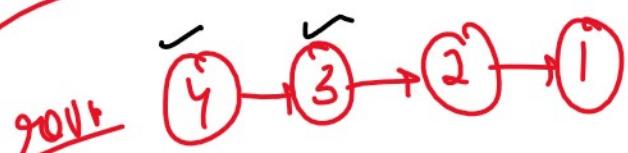
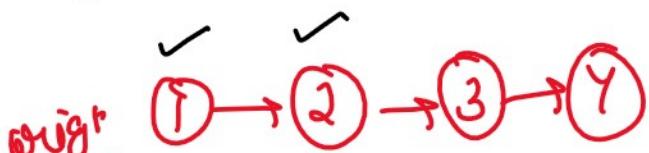
- 9 → UnorderedList, Stacks
- 10 → Stacks, Queues, bit man
- 11 → Trees (Binary)
- 12 → BST + Tries
- 13 → Heap, Graphs
- 14 → Graphs
- 15 → Greedy, DP?
- 16 → DP?

Reorder list

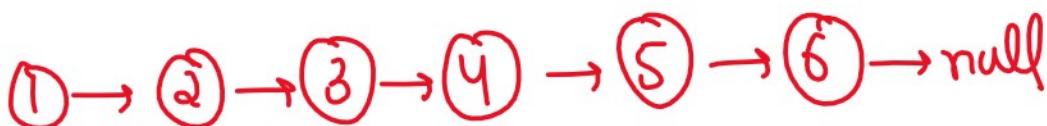
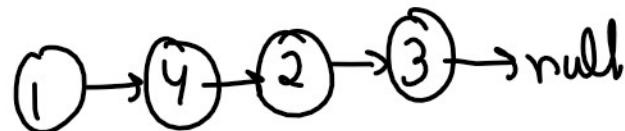
03 September 2023 13:12



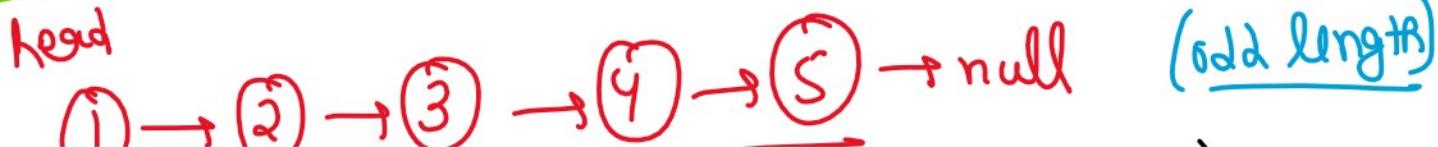
- ① reverse the list
- ② pick one from original and one from generated.



$\boxed{\begin{array}{l} \text{TC} = O(N) \\ \text{SC} = O(N) \end{array}}$ }
 $O(1)$



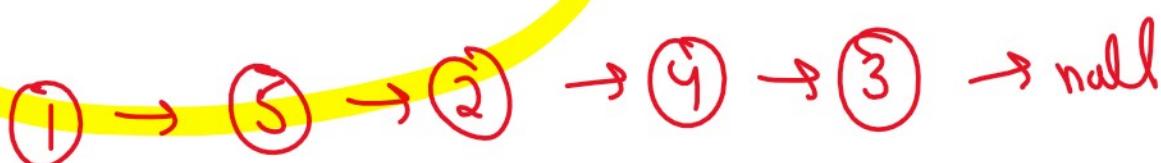
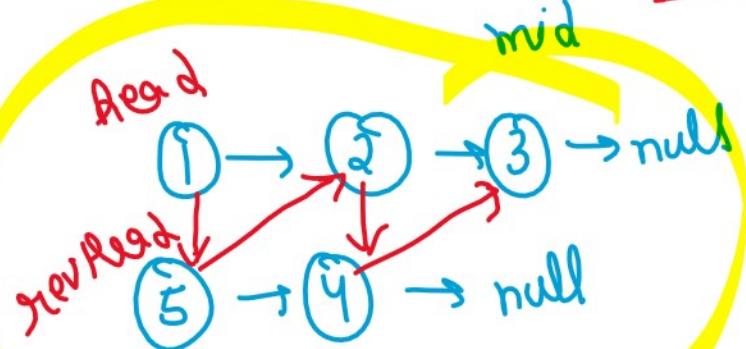
- ① find the mid point
- ② reverse the second half
- ③ Adjust the pointers





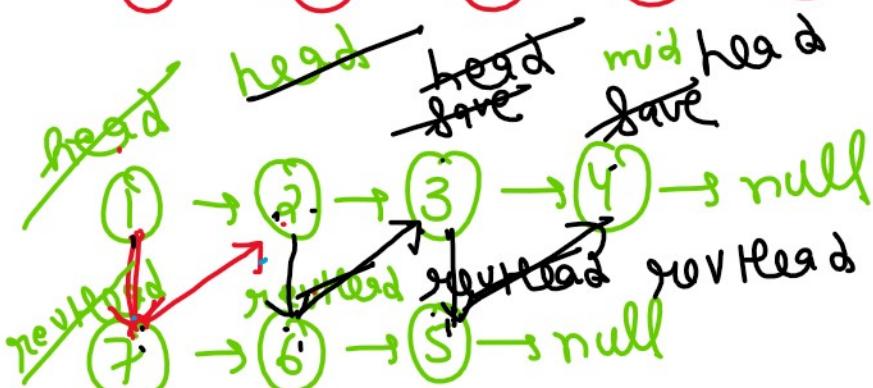
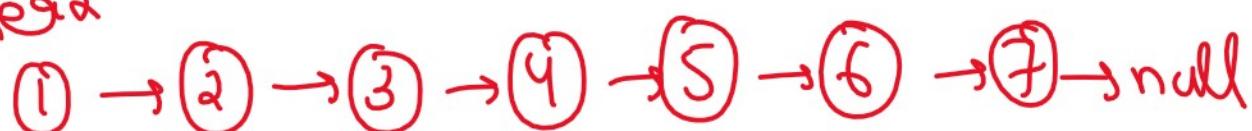
Time $\sim O(N)$

Space $\sim O(1)$



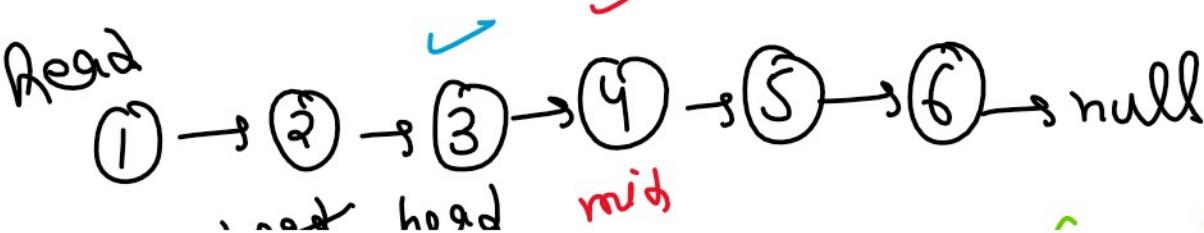
$$N + N + N = O(N)$$

head

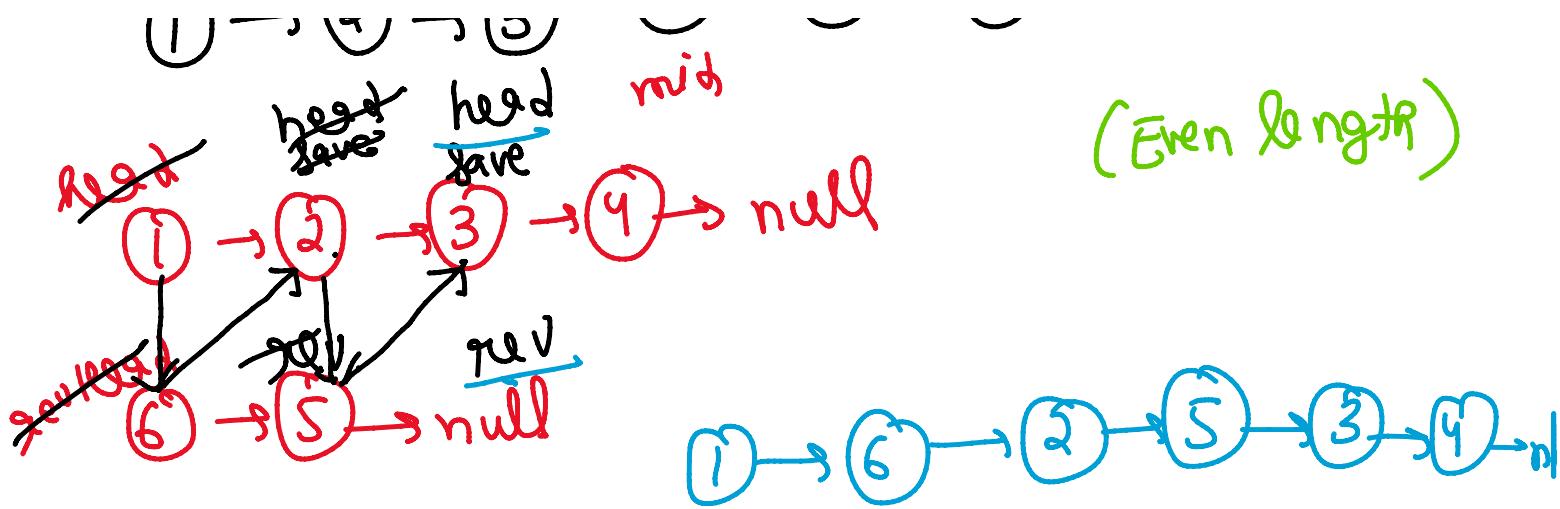


if $revHead \neq null$

- loop
1. save head's next node
 2. head \rightarrow next = revHead
 3. move revHead
 4. head \rightarrow next \rightarrow next
= save
 5. update head



~ n ... in \



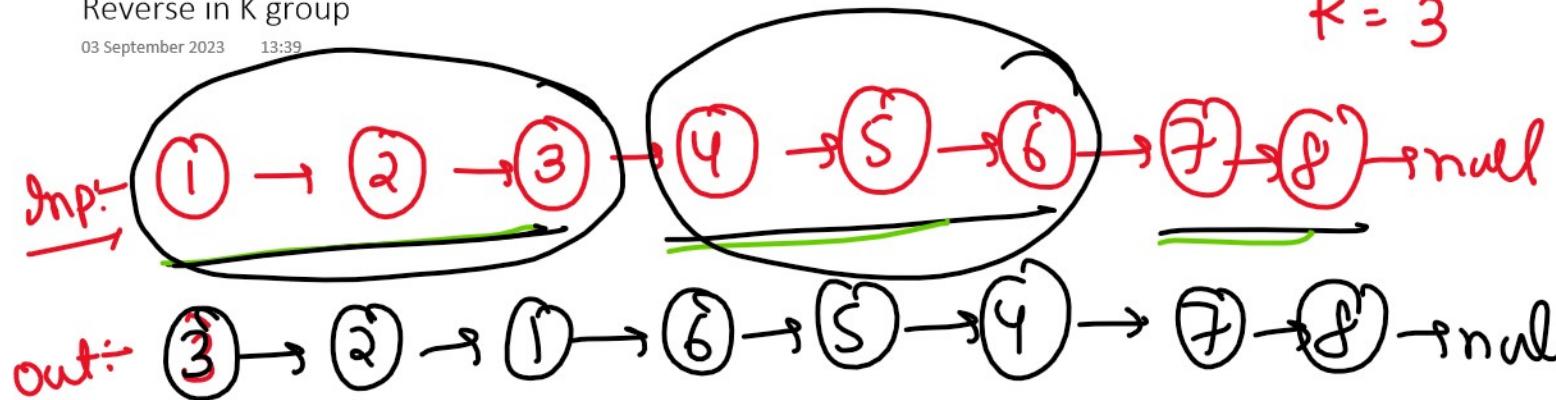
Reverse in K group

03 September 2023

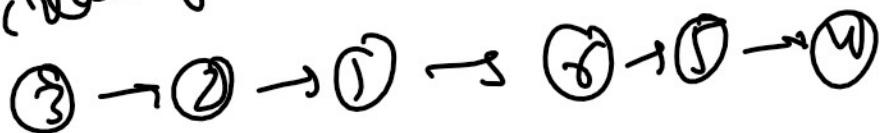
13:39

$$k = 3$$

Inp.



Getting new list

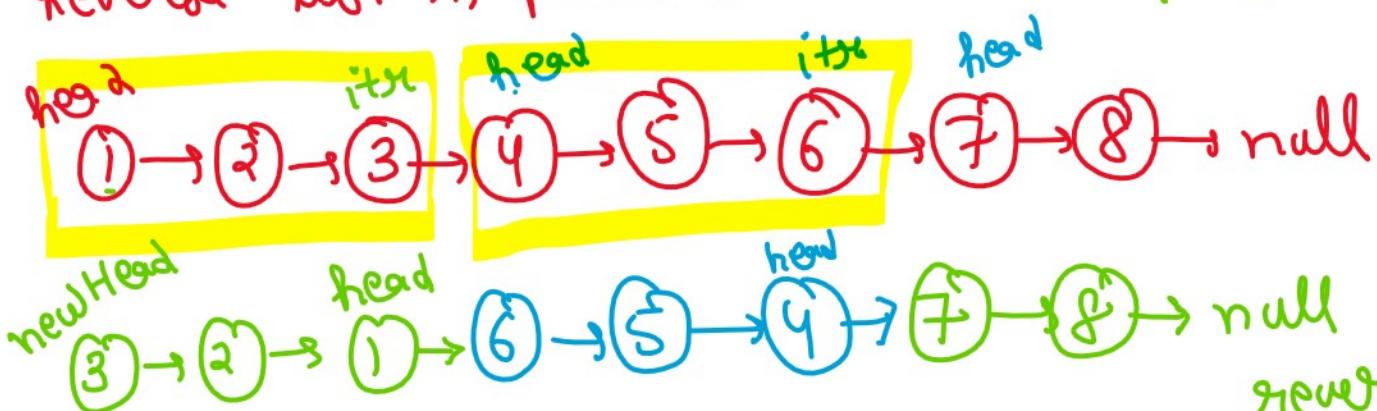


T. Rewardive and. Integrative

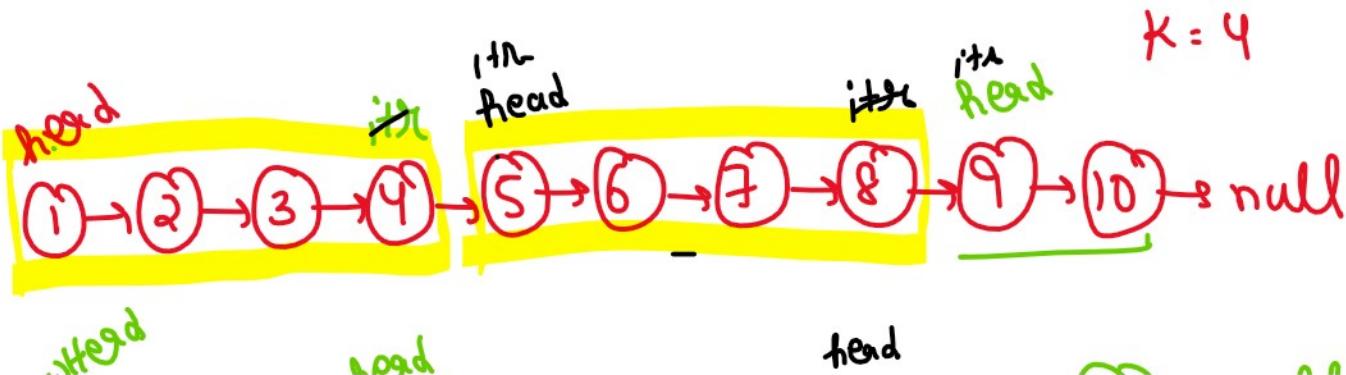
Time = $O(N)$
Space = $O(N)$

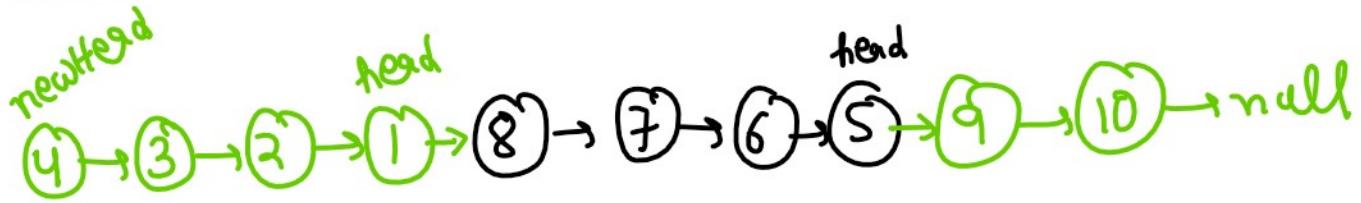
reverse list in pairs. ($K=2$)

$$k = 3$$



reverse(
head, iter)



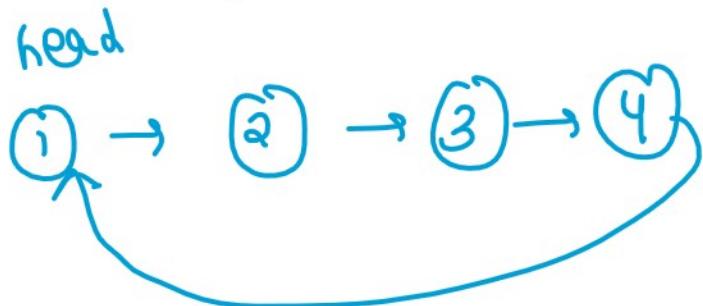


Time $\in O(N)$

$$\text{space} + \frac{N}{K} + 1 = O(N)$$

ignore space $+ O(1)$

- singly
- circular
- Doubly



HashMap + DLL

Stacks

03 September 2023 14:04

LIFO → last in first out

→ pile of books / plates

→ Box of CDs

→ Undo

→ Bangles in hand

→ Back operation in browser

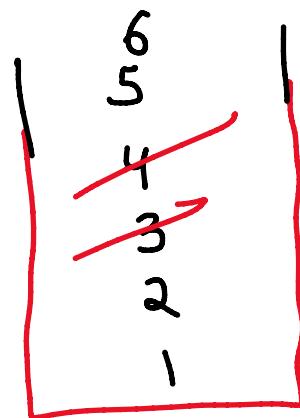
C++ (8+1)

push

pop

top

empty



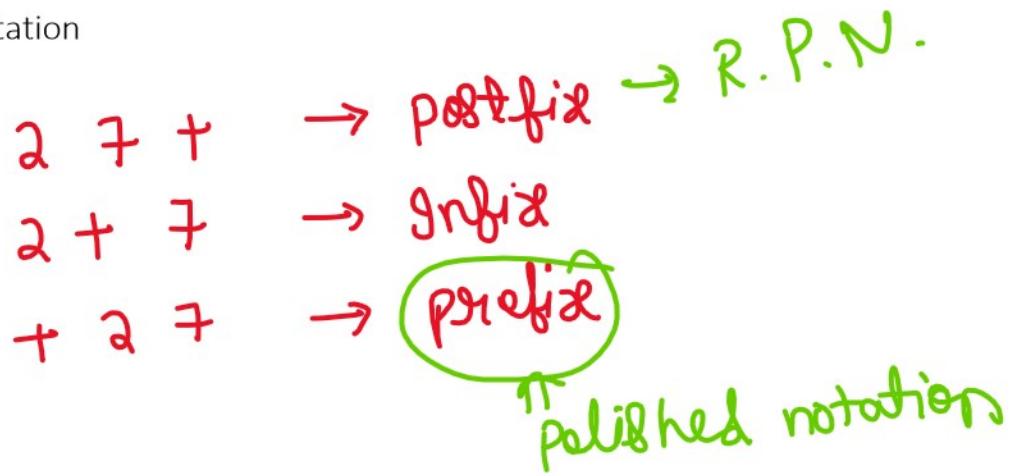
Java (Collections)

push
pop
peek
isEmpty

→ O(1)

Evaluate reverse polish notation

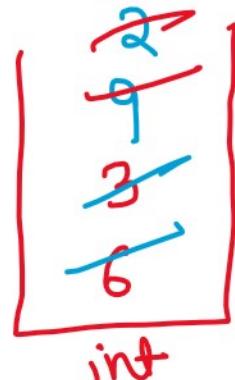
03 September 2023 14:13



$$\begin{aligned}
 \text{tokens} &= [5, 3, +, 2^*] & [8, 2, *] \\
 &= (5+3)^* 2 & = (18) \\
 &= (8)^* 2 = \underline{\underline{16}}
 \end{aligned}$$

$$\begin{aligned}
 \text{tokens} &= [5, 3, 2, +, ^*] & \text{stack?} \\
 &\quad \leftarrow & \\
 &= [5, 5, ^*] \\
 &= [25] = 25 \text{ and}
 \end{aligned}$$

$$\begin{aligned}
 \text{tokens} &= [6, 3, +, 2, ^*]
 \end{aligned}$$



$$\begin{aligned}
 x &= 3 \\
 y &= 6 \\
 \therefore x+y &= 3+6 = 9
 \end{aligned}$$

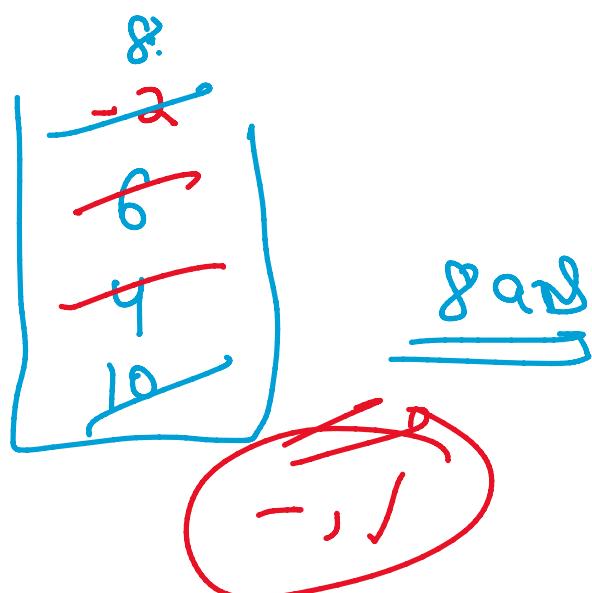
push the operand
in stack

stack.top();
| peek();

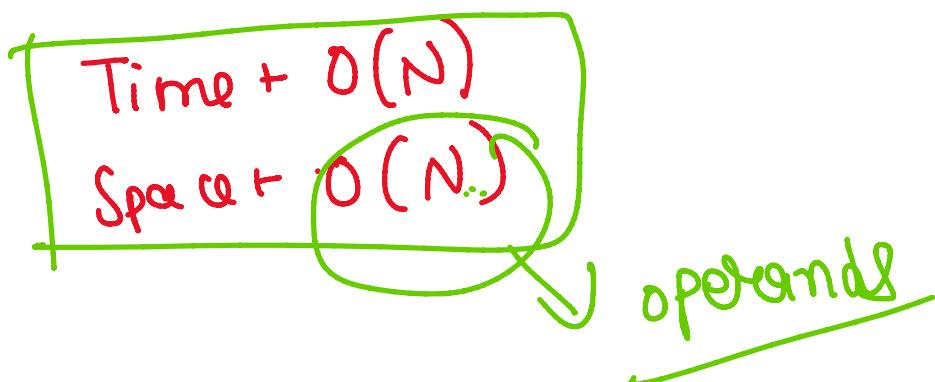
$y = 6$ |
 $x + y = 3 + 6 = 9$ | int | stack | peek();

tokens: $[\overline{10}, \overline{4}, \overline{6}, \overline{-}, \overline{+}]$

$$\begin{aligned} & \cancel{x = 6} - 2 \\ & y = 4 \quad \cancel{10} (y - x) \\ & (4 - 6) = -2 \end{aligned}$$



Ordering of pop will matter in case of sub, div.



Valid parentheses

03 September 2023 14:55

$() [\{]\}$ → invalid

$[()]\} \{$ → invalid

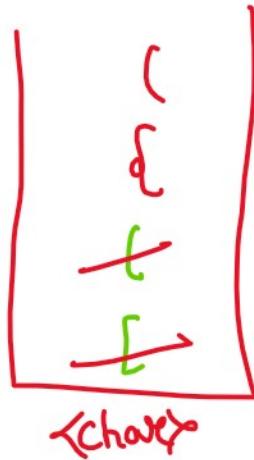
$[] \{ () \}$ → valid

→ $\cancel{(\}} \cancel{\{)} \cancel{(\}} \cancel{\{)}$ → false / invalid

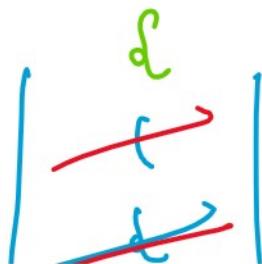
push the opening brackets in the stack.

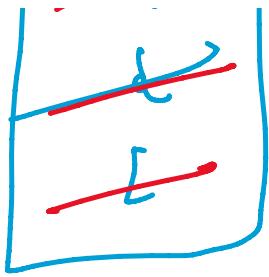
stack), The brackets which opens at last, will close first

false



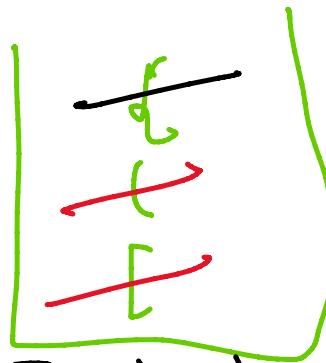
→ $\overline{[\{(\})\}} \overline{]} \overline{\{}$ → false





At last, stack should be empty.

- $\overbrace{[\{ \}] \{ \} }^s \cdot e \rightarrow \text{true}$



$\left\{ \begin{array}{l} \text{Time} + O(N) \\ \text{Space} + O(N) \end{array} \right.$

true ^{count of}
opening brackets

return stack.IsEmpty():

Daily temperatures

03 September 2023 15:18

Stock span problem
next greater element
next smaller element



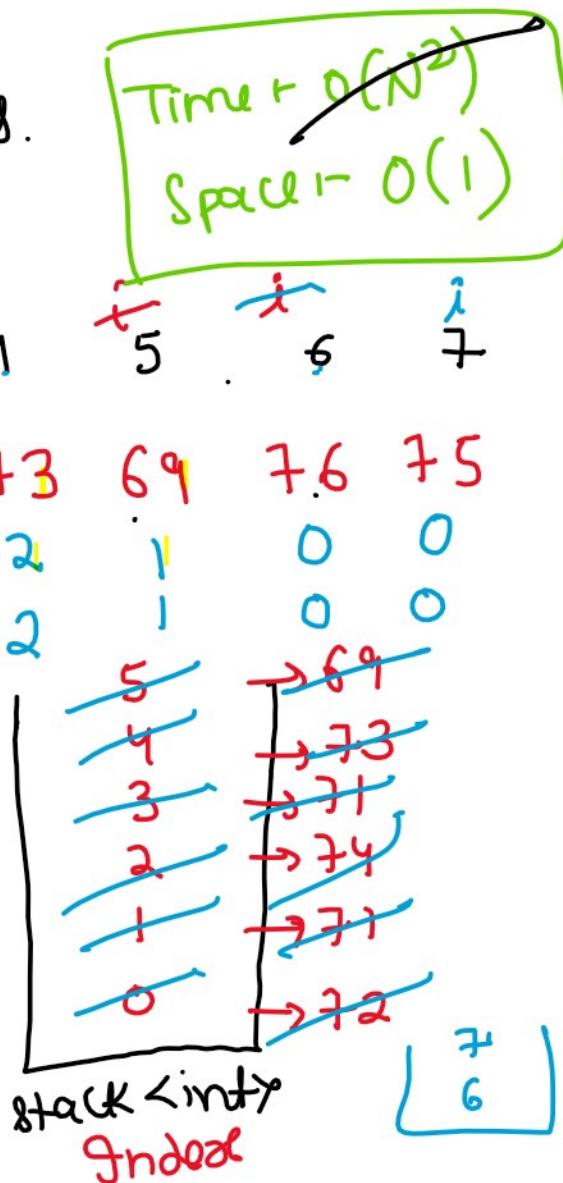
Brute force!. Two nested loops.

index	0	1	2	3	4	5	6	7
temp	72	70	74	71	73	69	76	75
ans	2	1	4	1	2	1	0	0
days	2	1	4	1	2	1	0	0

$$\text{days} = i - \text{s.top()}$$

(can we push the index?)

1. Push the lower temp.
2. If temp is warmer than calculate no. of days.



2. If today calculate no. of days. \Rightarrow stack $\sim n$ /
global

6

Time $\leftarrow O(N)$
Space $\leftarrow O(N)$

Min stack

03 September 2023 15:44

~~push(5)~~
~~push(7)~~
~~push(1)~~
~~-pop()~~

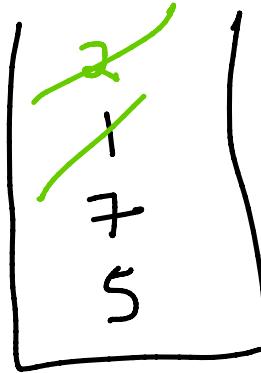
~~isEmpty()~~

~~push(2)~~

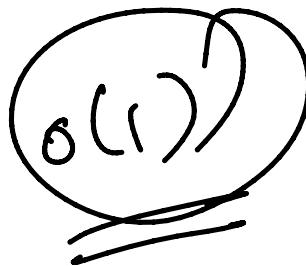
getmin() → 2

~~pop()~~

~~getmin()~~ → 5
~~top()~~ → 7



push
pop
top / peek
getmin



(create stack of type pair. pair<int, int> p = s.top();):

~~push(3)~~

~~push(7)~~

~~push(1)~~

~~getmin()~~ → 1

~~pop()~~

~~push(5)~~

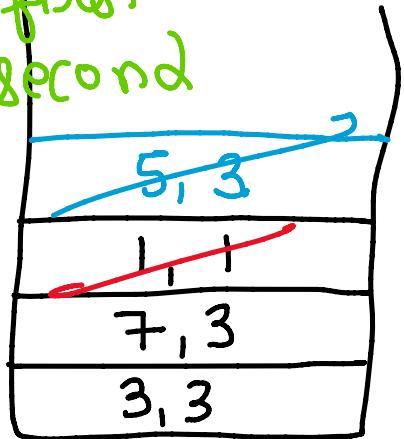
~~getmin()~~ → 3

~~peek()~~ → 5

~~pop()~~

~~getmin()~~ → 3

top = p.first
min = p.second

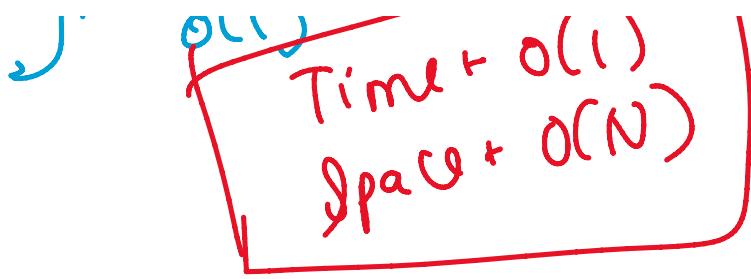


Stack<int, int>

s.top().second

O(1)
Time + O(1)

$\rightarrow \text{getmin}() \rightarrow 3$



wayback

Software Engineer II (L3) = 6+ years

Software Engineer III (L4) = 9-10+ years.