No. of valid Subarray :→

n = 5-0

n = 5

a b c d e   no. of Subarray = n

Subarray-

a
ab
abc
abcd
abcde

b
bc
bcd
bcde

c
cd
cde

d
de

e

2-1 = 1

	1	4	2	5	3	x
Index	0	1	2	3	4	5

~~1~~
~~1~~ 4
~~1~~ 4 2
~~1~~ 4 2 5
~~1~~ 4 2 5 3

~~4~~
× 4 2
× 4 2 5
× 4 2 5 3

~~2~~
~~2~~ 5
~~2~~ 5 3

~~5~~
~~5~~ 3

~~3~~

→ 11 Ans

Starting Element is smaller-

Index →	0	1	2	3	4
known array →	1	4	2	5	3.
next smaller Index on Right →	*5	2	*5	4	*5

contribution in count → 5 + 1 + 3 + 1 + 1

count += nsr[i] - i

Hint :→ Next Smaller on Right.

1
1 4
1 4 2
1 4 2 5
1 4 2 5 3

4

2
2 5
2 5 3

5 , 3

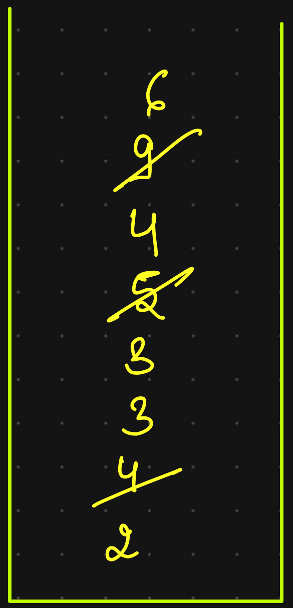
lexicographically Smallest Subseq:

$\boxed{k=4}$ → 4 length Smallest lexicographical subseq.

$$\underbrace{\underline{2}\quad \underline{4}\quad \underline{3}\quad \underline{3}\quad \underline{5}\quad \underline{4}\quad \underline{9}\quad \underline{6}}$$

k Remove → Removal = nums.length - k

= 6-4 = 4



$k = \cancel{4}\ \cancel{3}\ \cancel{2}\ 1$

$\underbrace{2\ \ 3\ \ 3\ \ 4\ \ \cancel{6}}$ ⇒ [2, 3, 3, 4]

Design a stack with Increment operation:

complexity → push → O(1)

pop → O(1)

increment → O(1)

push

pop

increment(k, val)

3, 4

10

2

4

3

1

x 7 * Nothing to do
└ stack is full

pop → 1

pop → 3

3, 2

1

20, 2

pop → 8

pop → 8

1

1

2, 6

pop → 7

pop → 7

pop → 12

pop → 20

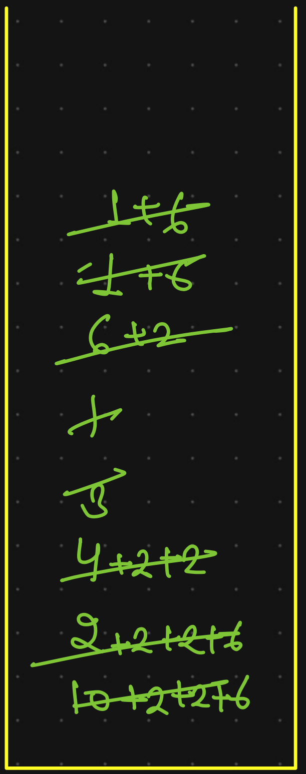
pop → -1

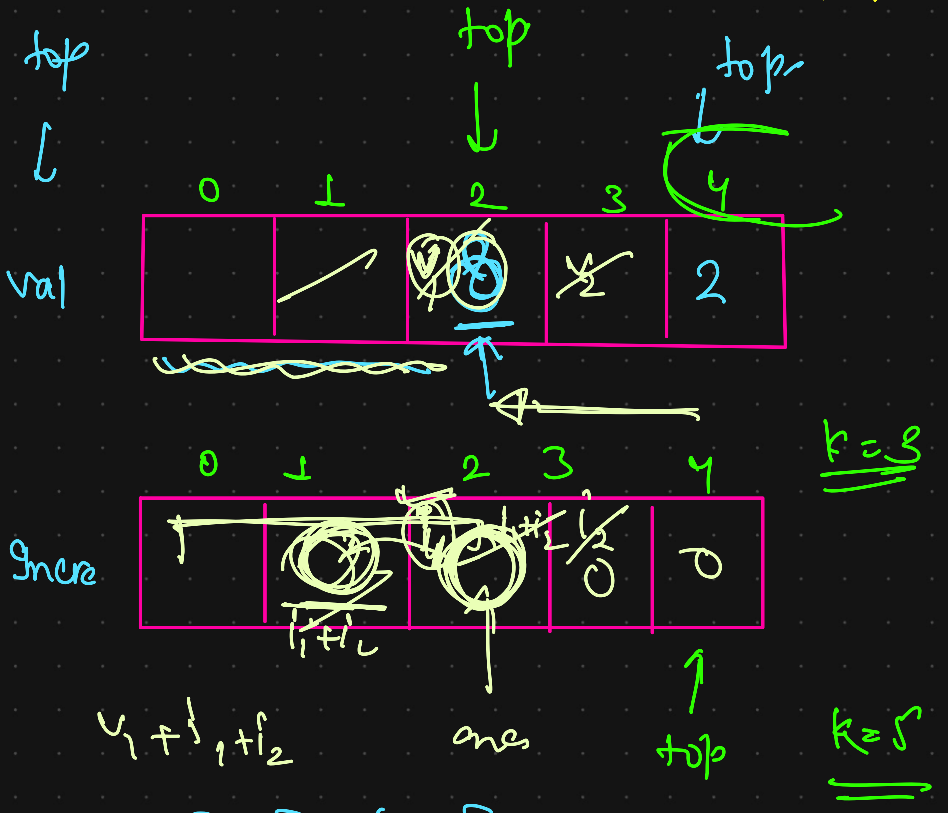
pop → -1

pop → -1

stack is empty.

maxsize = 5





pop → value = val[top] + incr[top];

v_2+i_2

incr[top-1] += incr[top]

top--; → check for valid index