

No. of valid Subarray: \rightarrow

$$n = 5 - 0$$
$$n = 5$$

a b c d e

no. of subarray = n

Subarray-

<u>a</u>	b	c	d	e
<u>a</u> b	b <u>c</u>	c <u>d</u>	d <u>e</u>	
a <u>b c</u>	b <u>c d</u>	c <u>d e</u>		
a <u>b c d</u>	b <u>c d e</u>			
a <u>b c d e</u>				

$2 \rightarrow 1 = 1$
 ① ④ 2 5 3 x
 index 0 1 2 3 4 5

Handwritten notes showing the steps of bubble sort on the array [1, 4, 2, 5, 3]. The array is written in a purple oval, and comparisons are shown in teal ovals. Red lines indicate swaps. The process shows the first pass where 1 and 4 are compared, 4 and 2 are compared and swapped, 2 and 5 are compared, and 5 and 3 are compared and swapped. The second pass shows 1 and 4 are compared, 4 and 2 are compared and swapped, and 2 and 5 are compared. The third pass shows 1 and 4 are compared, and 4 and 2 are compared and swapped. The final array is 1, 2, 4, 5, 3.

→ 11 Aug

starting element is smaller-
index $\rightarrow 0, 1, 2$

known array →

Index →	0	1	2	3	4
	1	4	2	5	2

next smaller
Index on Right \rightarrow 5 2 5 4 5

Contribution in cost $\rightarrow \underbrace{5 + 1 + 3 + 1 + 1}_{\text{5}}$

count += 1 nst[i] = i

$$\begin{array}{r} 1 \\ 24 \\ 425 \\ 4253 \end{array} \quad \begin{array}{r} 4 \\ 25 \\ 253 \end{array} \quad \begin{array}{r} 5 \\ 53 \end{array}$$

lexicographically Smallest Subseq^o

2 4 3 3 5 4 9 6

k=4

→ 4 length Smallest
lexicographical subseq.

k Remove

$$\begin{aligned} \text{Removal} &= \text{nums.length} - k \\ &= 6 - 4 = \textcircled{4} \end{aligned}$$

2
4
3
3
5
4
9
6

k=4 3 2 1

3
2
1

2 3 3 4 ~~6~~

⇒

[2, 3, 3, 4]

Design a stack with increment operation

complexity \rightarrow push $\rightarrow O(1)$

pop $\rightarrow O(1)$

increment $\rightarrow O(1)$

push

pop

increment(k, val)

~~10~~

~~2~~

~~4~~

~~3~~

~~1~~

~~7~~ * Nothing

pop \rightarrow 1 to 80
stack is full

pop \rightarrow 3

~~3, 2~~

~~6~~

~~20, 2~~

pop \rightarrow 8

pop \rightarrow 8

~~1~~

~~1~~

~~20, 6~~

pop \rightarrow 7

pop \rightarrow 7

pop \rightarrow 12

pop \rightarrow 20

pop \rightarrow -1

pop \rightarrow -4

pop \rightarrow -1

stack is empty

maxsize = 5

~~1+6~~
~~1+6~~
~~6+2~~
+
3
~~4+2+2~~
~~2+2+2+6~~
~~10+2+2+6~~

top

val

incr

3, 4

top

top

0	1	2	3	4
		8	12	2

0	1	2	3	4
	1+1	1+1	1+1	0

$v_1 + i_1 + i_2$

one

top

k=8

k=5

pop \rightarrow value = val[top] + incr[top]; $v_2 + i_2$

incr[top-1] = incr[top]

top--: check for valid index