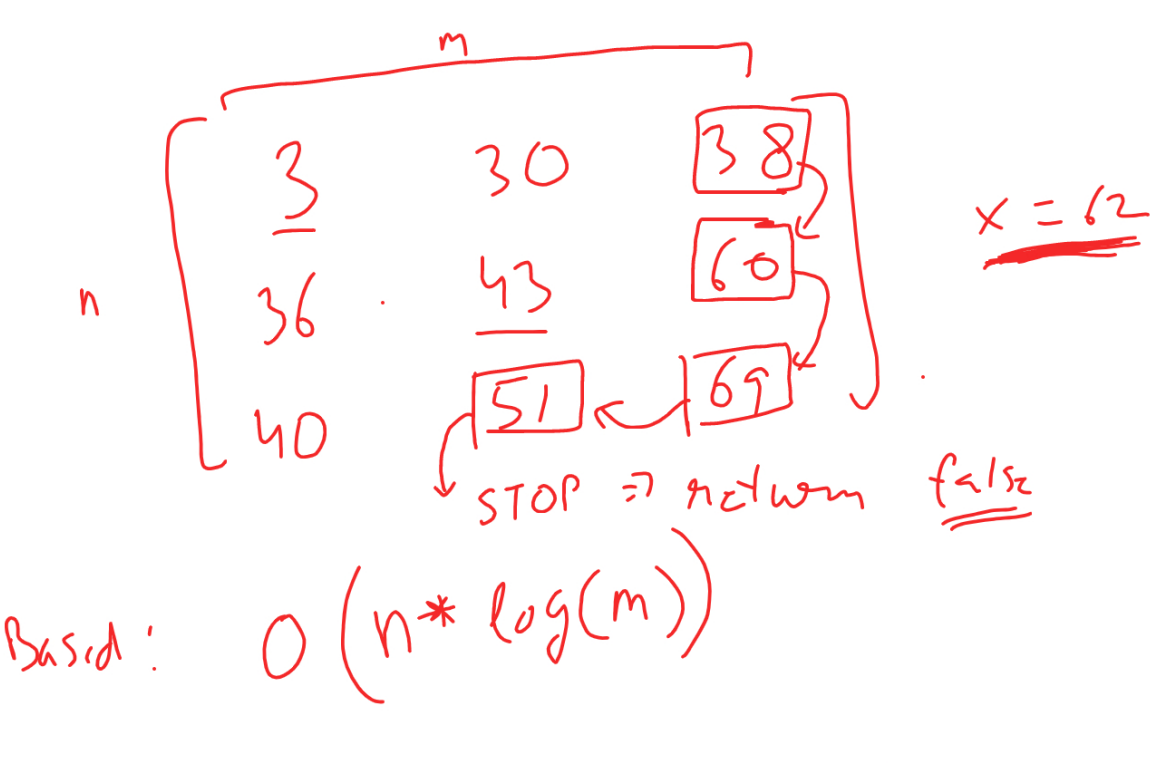
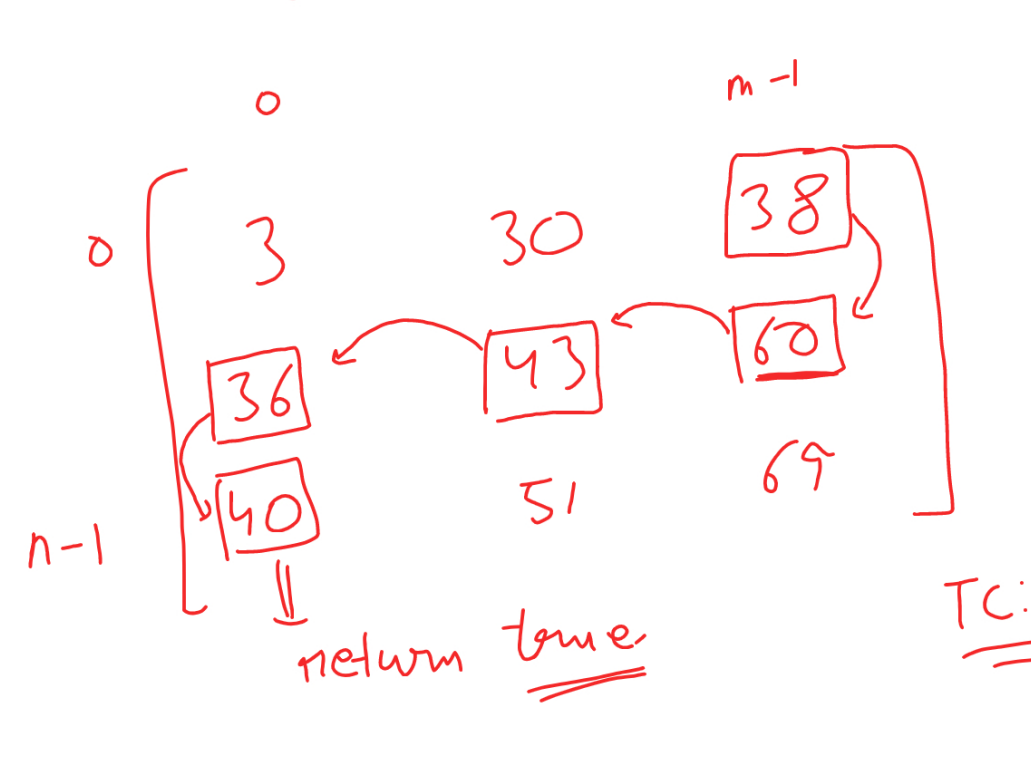

m
n
3 30 38
36 43 60
40 51 69
STOP ⇒ return false

x = 62

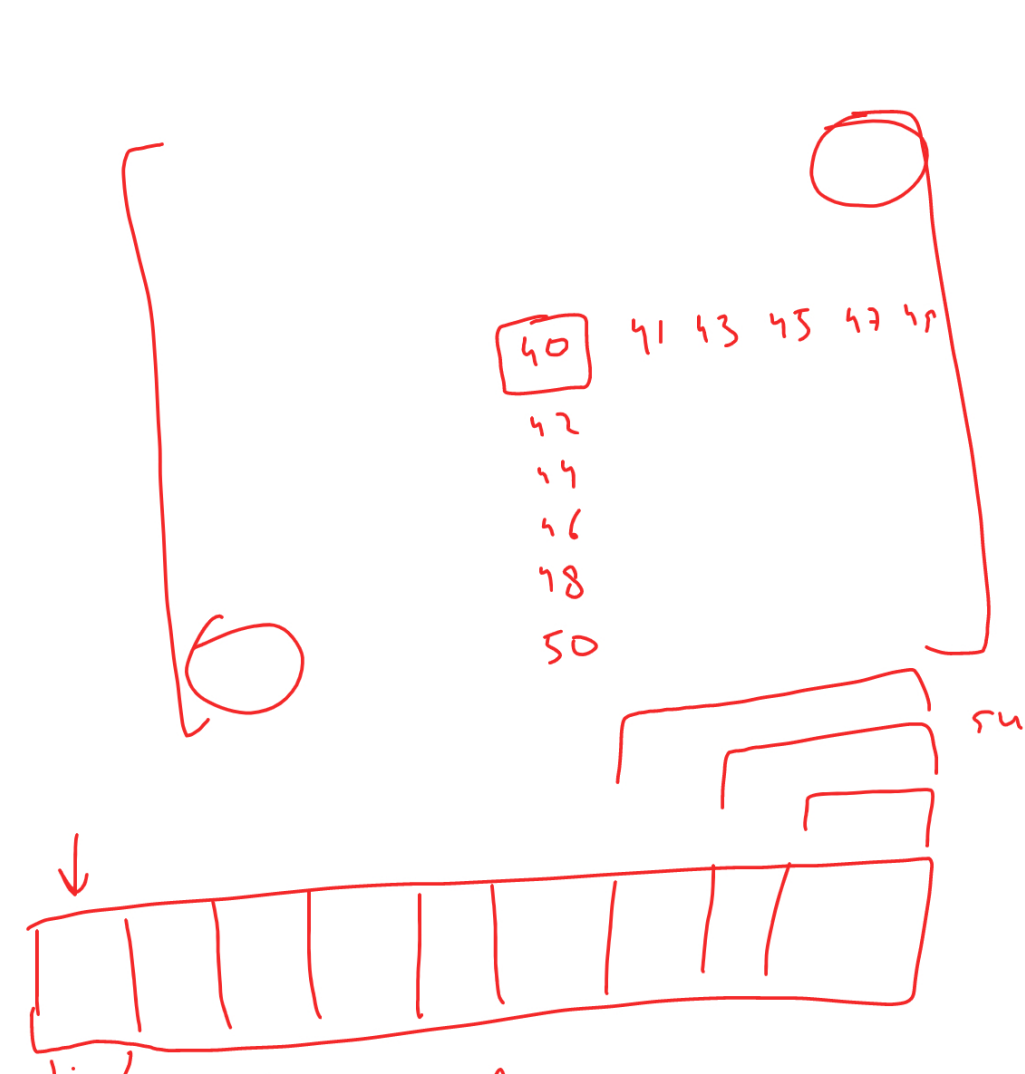
O(n*m)

BS-Based: O(n* log(m))


0
m-1
0
n-1
3 30 38
36 43 60
40 51 69
return true

x = 40

TC: O(n+m)


40 41 43 45 47 49
42
44
46
48
50
suffix
prefix

10×10

48

[Prefix → Any subarray starting from the beginning.]

[Suffix → Any subarray ending at the end]

1, 9

L[] = [2 1 3]
R[] = [5 3 9]

[2, 5] : [2, 3, 4, 5]
[1, 3] : [1, 2, 3]
[3, 9] : [3, 4, 5, 6, 7, 8, 9]

element	freq
1	1
2	2
⇒ 3	3
4	2
5	~~3~~ 2
6	1
7	1
8	1
9	1

[2, 5] : [2, 3, 4, 5]
[1, 4] : [1, 2, 3, 4]
[3, 9] : [3, 4, 5, 6, 7, 8, 9]

elt	freq
3	3
4	3

[0 , 0 , 0]
[10^5 , 10^5 , 10^5]

Hashmap-based approach.

TC: O(n * max_range_length)

WC: n = 10^6, L[] = [0, 0, 0 . . . . 0]
R[] = [10^6, 10^6, 10^6 . . . . 10^6]

[max no. of computations = 10^6 × 10^6 = $\boxed{10^{12}}$]

$\boxed{10^8}$ ⇒ max LIMIT

n = 10^5 $\boxed{O(n^2)}$

[L[] = [1, 2, 3, 100]
R[] = [5, 5, 5, 100]]

[0, 10^6]

L[] = [2 1 3]
R[] = [5 3 9]

	0	1	2	3	4	5	6	7	8	9	10	11
freq:	0	1	1	1	-1	0	-1	0	0	0	-1	0
cum. freq:	0	1	2	3	2	2	1	1	1	1	0	0

[for L[i] → increment 1 at index L[i]
R[i] → decrement 1 at index R[i]+1]
L[i] → R[i]

cum_freq[i] = freq[0] + freq[1] - - - - + freq[i]

[cum_freq[i] == frequency of element 'i' in all of the ranges.]

L[] = [3]
R[] = [7]

8

⑦ 8.

size = 9.

	0	1	2	3	4	5	6	7	8
freq:	0	0	0	~~0~~ +1	0	0	0	0	~~0~~ -1
cum-freq:	0	0	0	1	1	1	1	1	0

TC: O(n + max_range)

Aux-space: O(max_range)

Space Comp: O(n + max_range)

[prefix_sum[i] = prefix_sum[i-1] + current_element]

	0	1	2	3	4	5	6	7	8	9	10	11
freq:	0	1	1	1	-1	0	-1	0	0	0	-1	0
cum. freq:	0	1	2	3	2	2	1	1	1	1	0	0

[result = ~~2~~ ~~2~~ 3]