HashSet<Integer> set = new HashSet<>()

→ no duplicate value
→ get in O(1)
→ order is not preserved

Ques Find Common Elements

⇒ 3, 4, 1

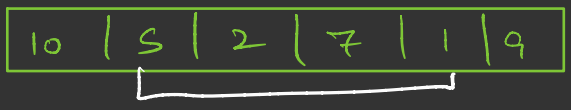
Brute Force

For each elem in arr1, we will search the elem in arr2

⇒ O(n × m) ⇒ O(n²)

O(n)
+
O(n) → O(n)

Q.ees Length of longest Subarray with sum = k



k = 15

maxL = 4

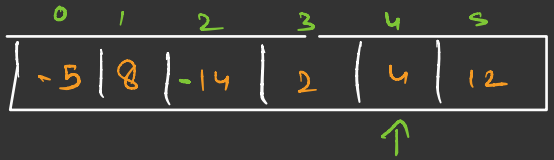
Brute Force

for (start → end)

for (start to end)
→ n

O(n2)

k = -5



max = 4

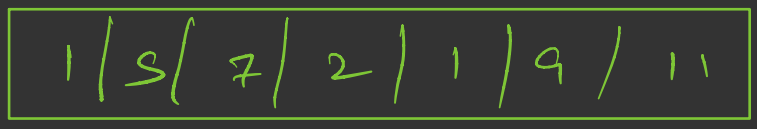
ps	first ind
0	-1
-5	0
3	1
-11	2
-9	3
4	4

ps -5 3 -11 -9 -

ps - k

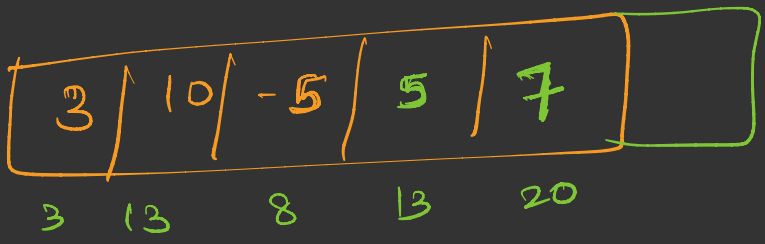
-5 - (-5)

-10



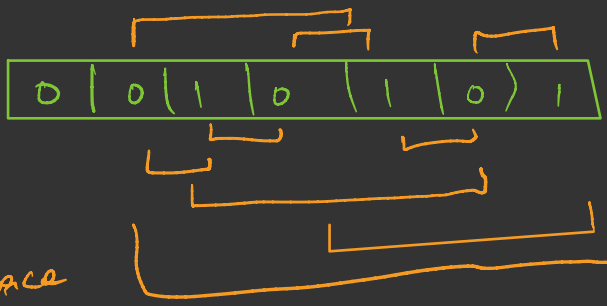
bs - k
16 - 15
⇒ 1

0	-1
1	0
6	1
13	2
15	3
16	4
25	5



k = 7

Ques



⇒ 9

Brute Force — $O(n^2)$

Count all sub arrays with equal 0 and 1

c0 = ~~12~~
c1 = ~~0~~ x 2

total count = ~~0~~ x 2

ps -1 -2 -1 -2 -1 -2 -1

ps	Count
0	0
-1	1
-2	2
	3

Count = 0 1 2 3 4 5