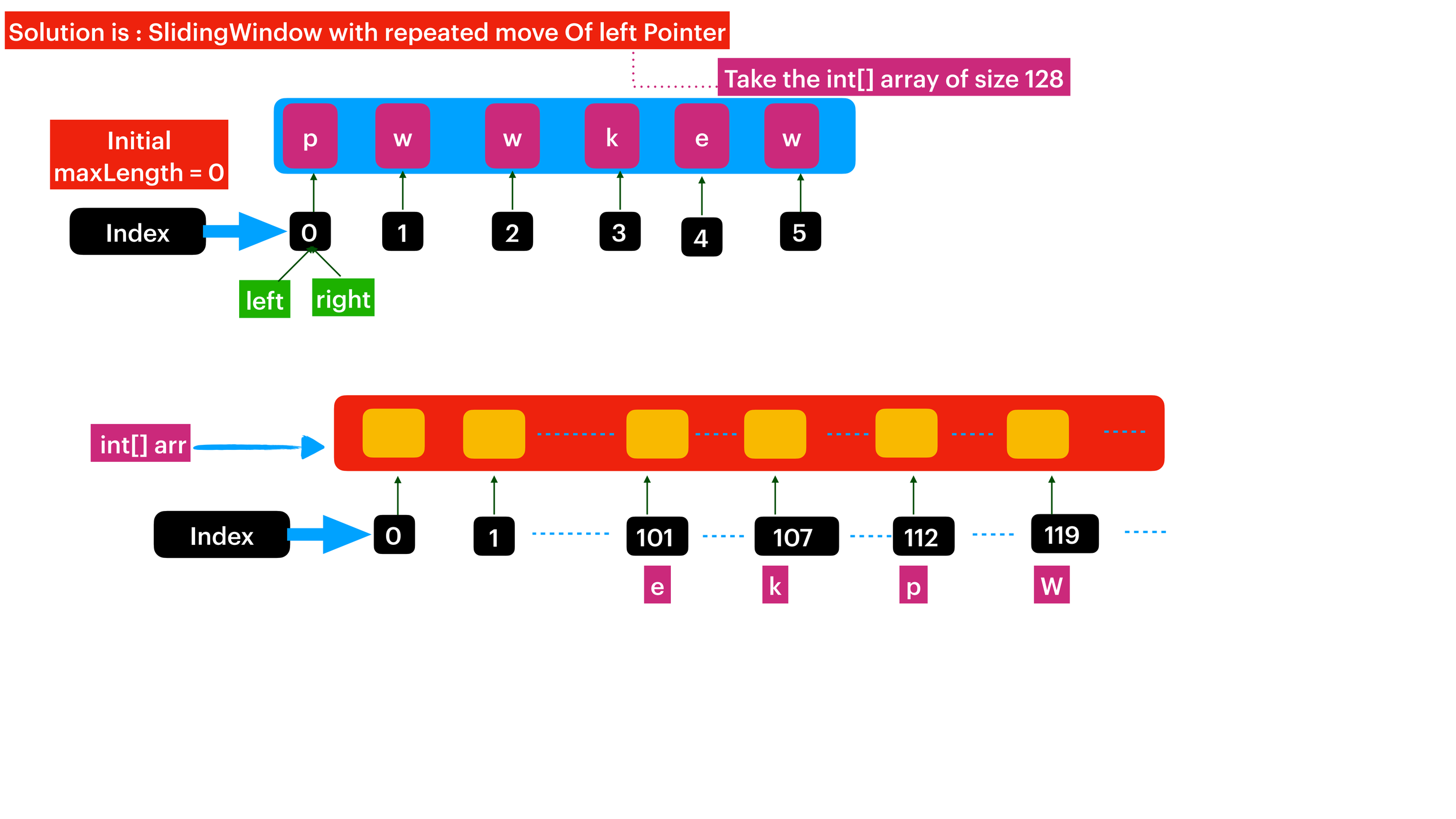
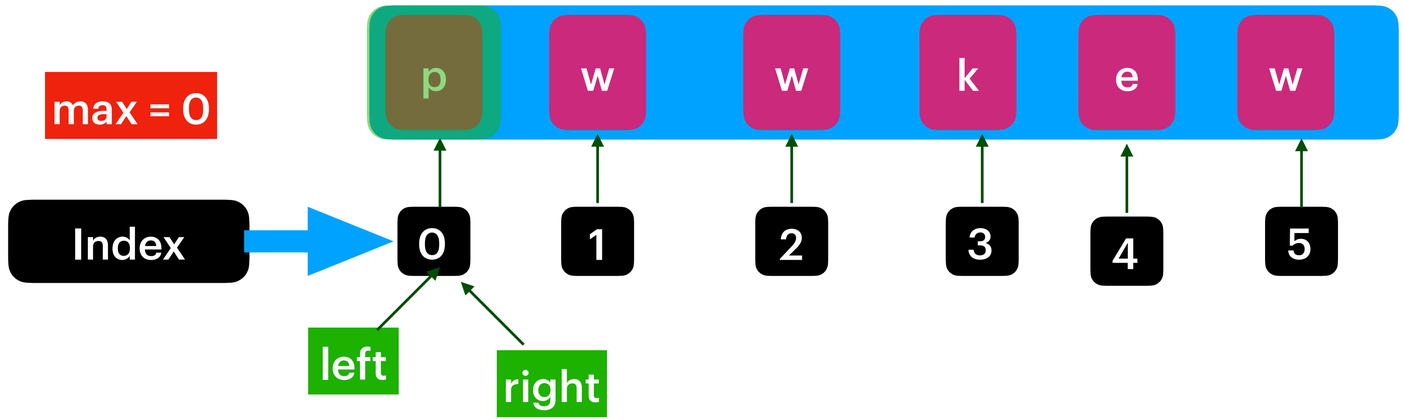
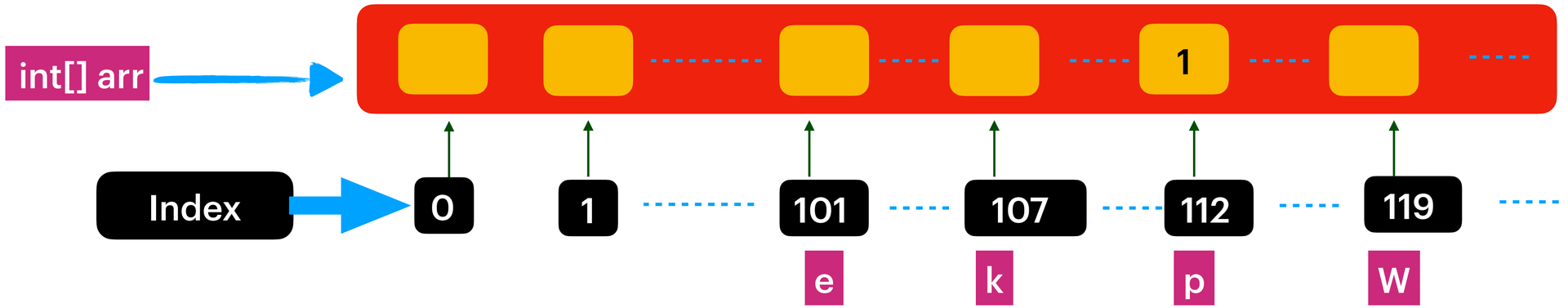

Solution is : SlidingWindow with repeated move Of left Pointer
Take the int[] array of size 128
Initial
maxLength = 0
p
w
w
k
e
w
Index
0
1
2
3
4
5
left
right
int[] arr
Index
0
1
101
107
112
119
e
k
p
w

Solution is : SlidingWindow with
repeated move of leftPointer when there is duplicate character

Take the int[] array of size 128
Represents all the characters ASCII values



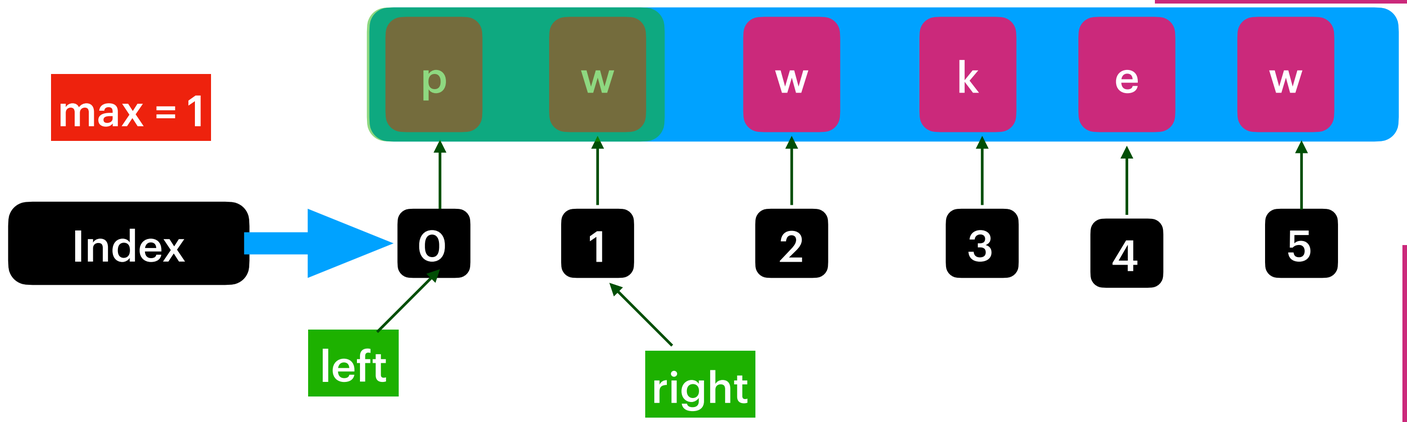

arr[right] = arr[right] + 1
arr[p] = arr[p] + 1
arr[112] = arr[112]+1 = 0 + 1 = 1



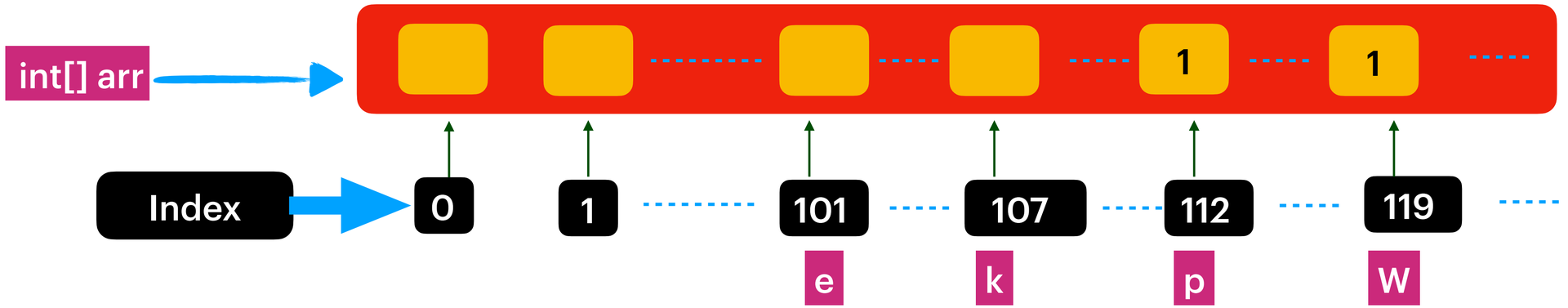

As arr[112] is not > 1 so move right Pointer.
max = Math.max(max, right-left+1)
max = Math.max(0,0-0+1) = 1

Solution is : SlidingWindow with
repeated move of leftPointer when there is duplicate character

Take the int[] array of size 128



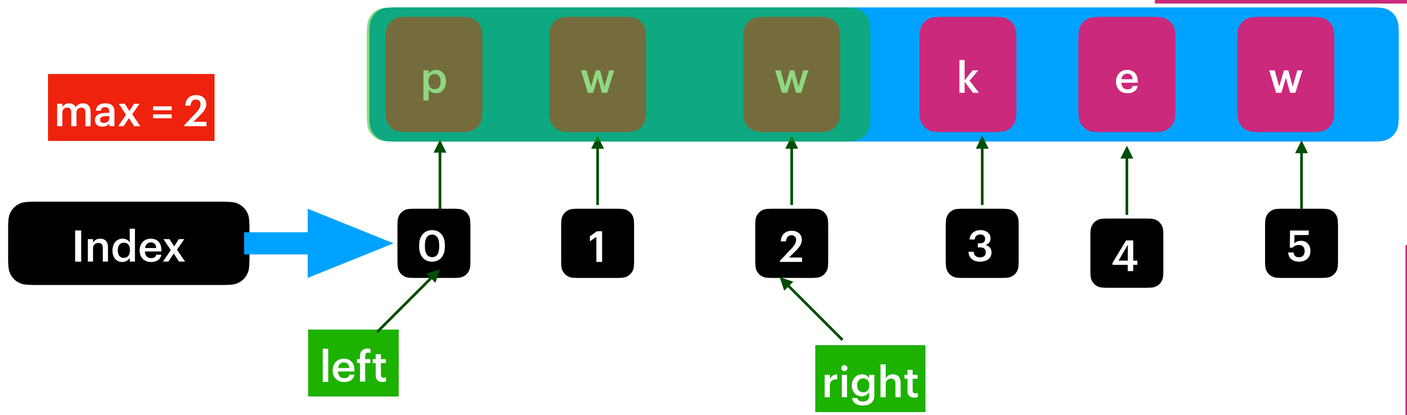

arr[right] = arr[right] + 1
arr[w] = arr[w] + 1
arr[119] = arr[119]+1 = 0 + 1 = 1



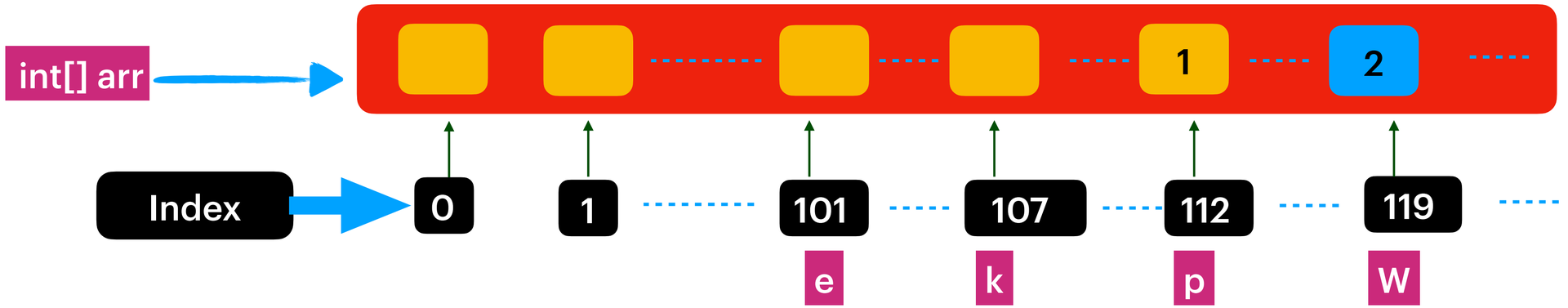

As arr[119] is not > 1 so move rightIndex
max = Math.max(max,right-left+1)
=Math.max(1, 1-0+1) = 2

Solution is : SlidingWindow with
repeated move of leftPointer when there is duplicate character

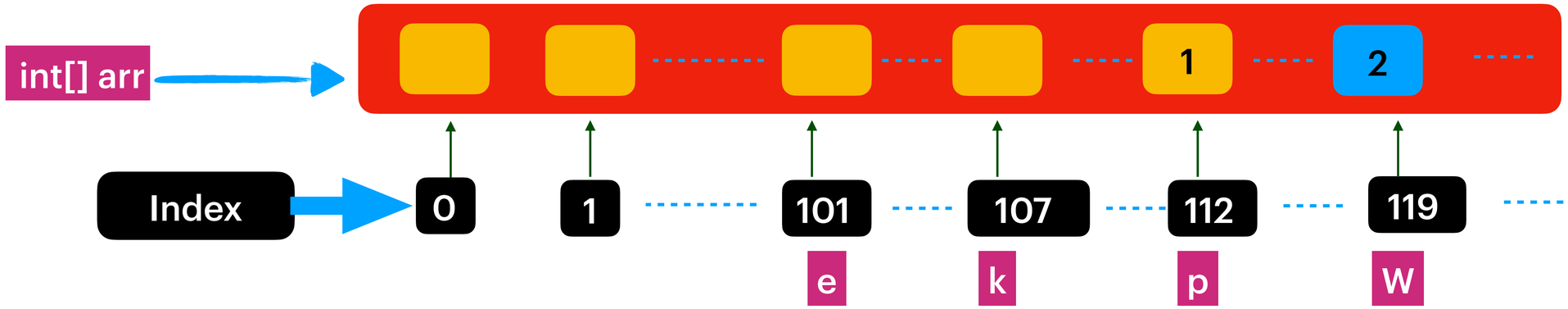
Take the int[] array of size 128

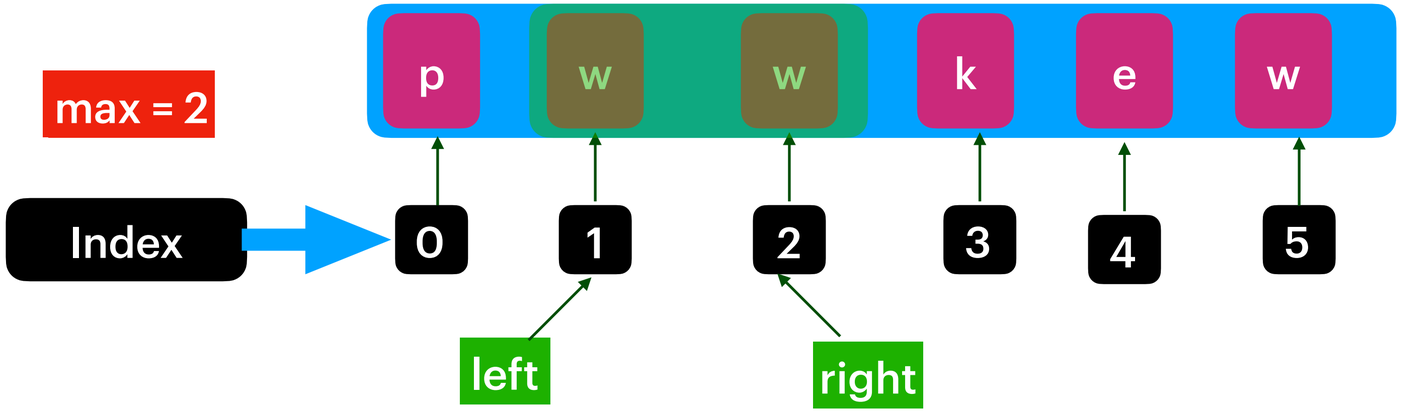



arr[right] = arr[right] + 1
arr[w] = arr[w] + 1
arr[119] = arr[119]+1 = 1 + 1 = 2

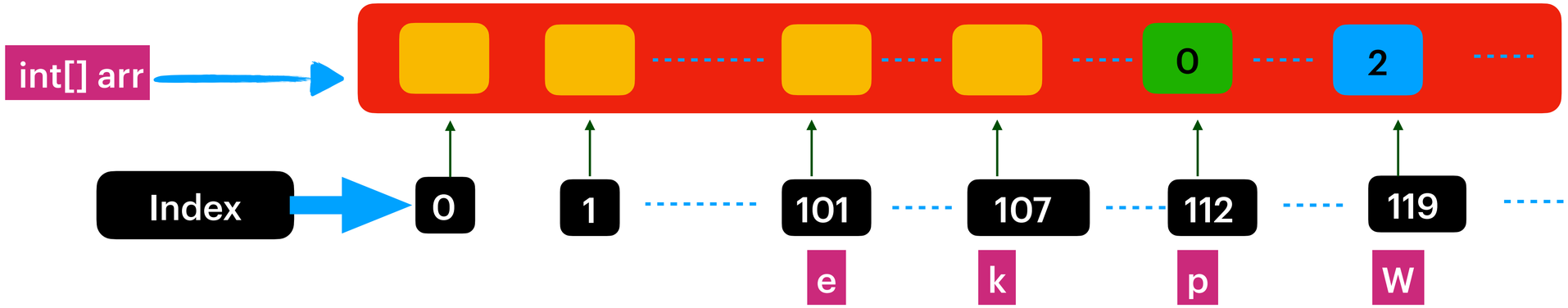


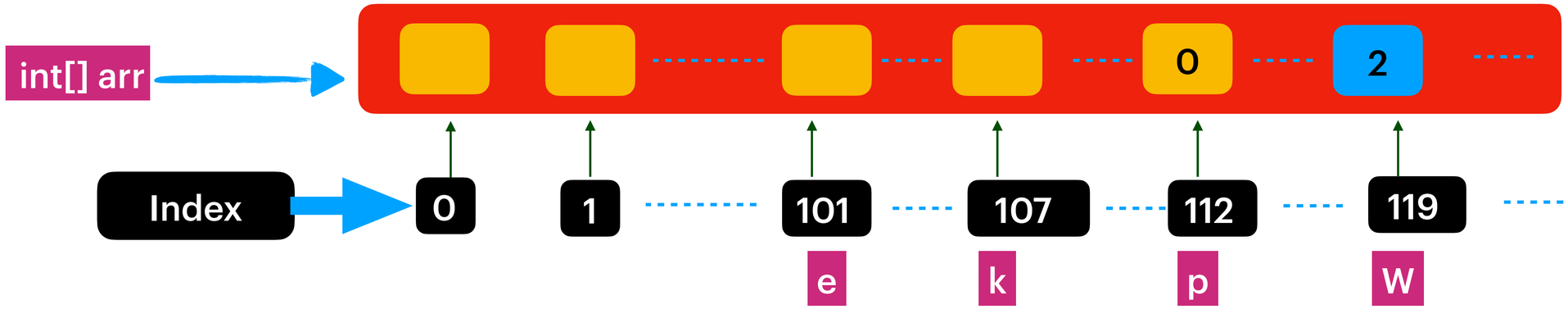

As arr[119(right)] is > 1 Repeat below process till arr[119] <= 1
Decrement current left Pointer value
forward the leftPointer to one index forward

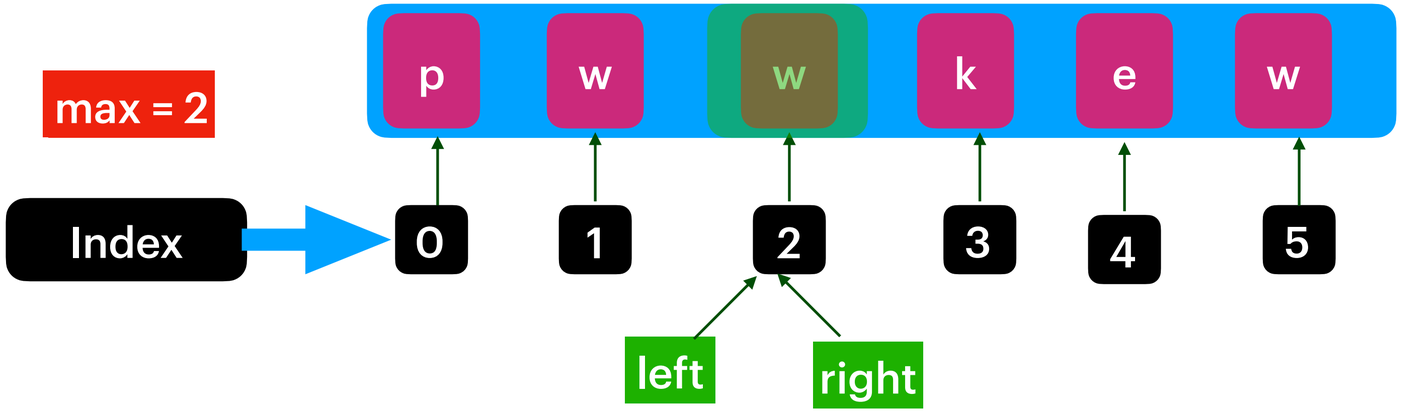


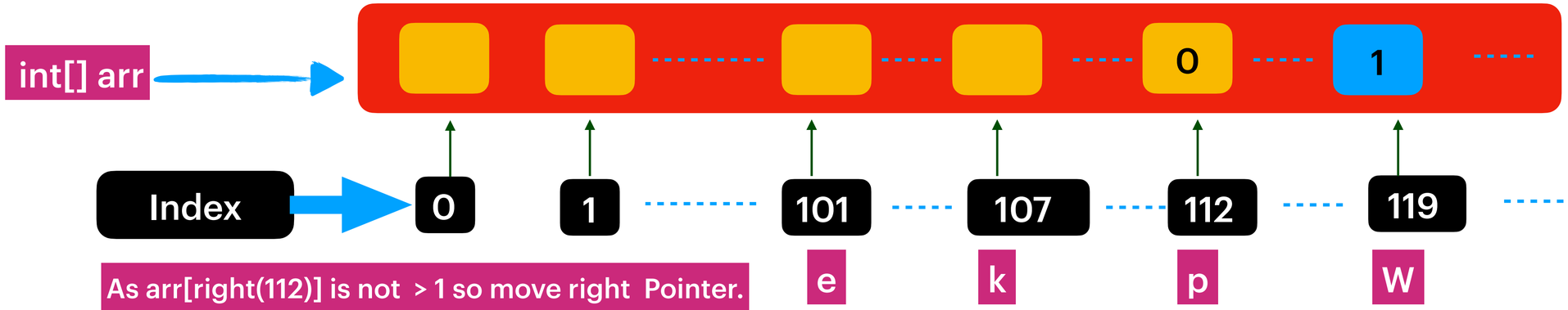

As arr[(right)119] is > 1 so move rightIndex :
Repeat below process till arr[119] <= 1
Decrement current right index value
move leftPointer to one
index forward .
update the max = Math.max(max, right-left+1)
max = Math.max(0,2-1+1) = 2





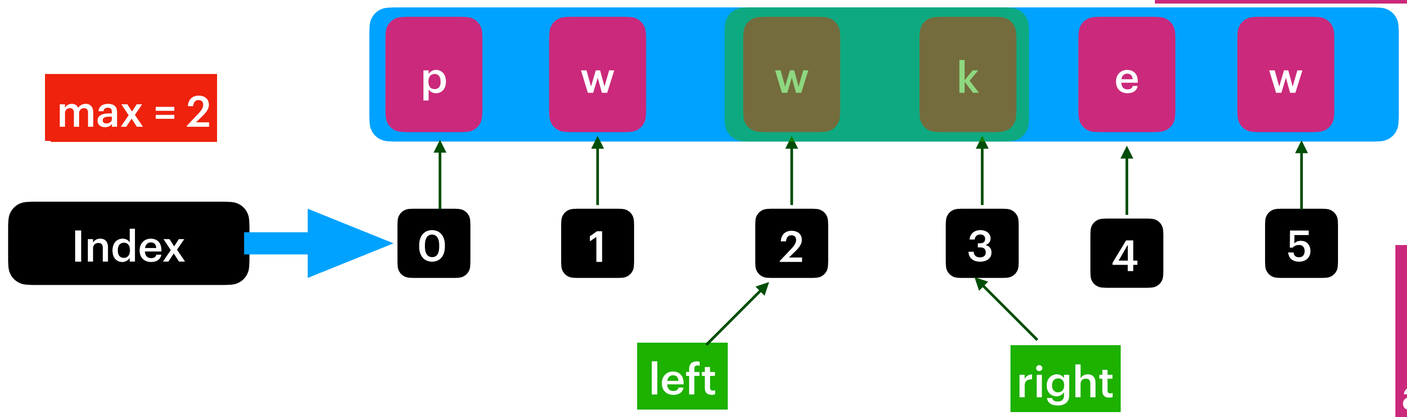

As arr[119] is > 1 so move rightIndex :
Repeat below process till arr[119] <= 1
Decrement current right index value
move leftPointer to one
index forward .
update the max = Math.max(max, right-left+1)
max = Math.max(0,2-1+1) = 2



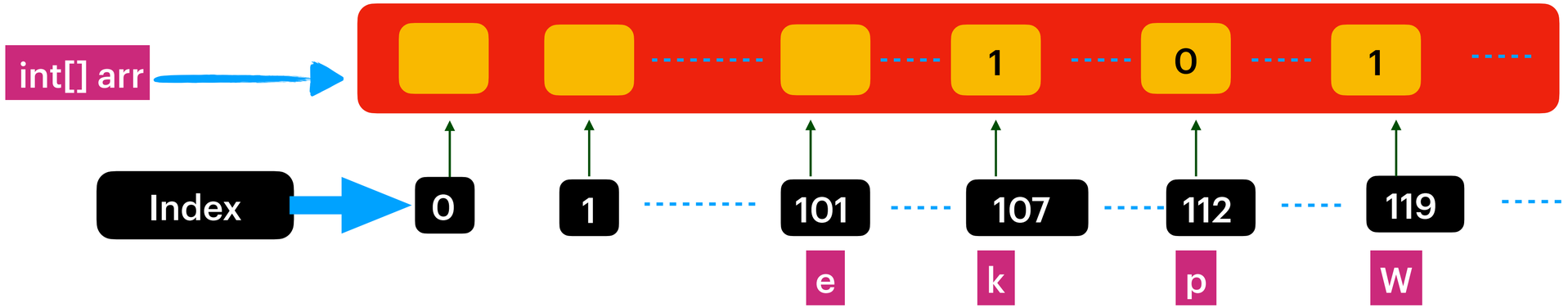

As arr[right(112)] is not > 1 so move right Pointer.
max = Math.max(max, right-left+1)
max = Math.max(2, 2-2+1) = 2

Solution is : SlidingWindow with
repeated move of leftPointer when there is duplicate character

Take the int[] array of size 128



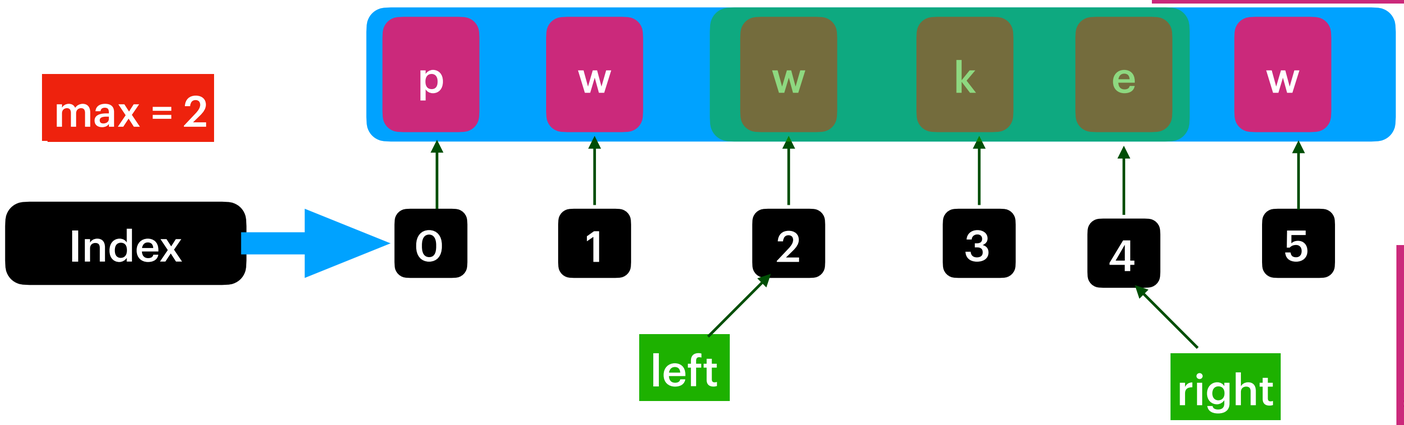

arr[right] = arr[right] + 1
arr[k] = arr[k] + 1
arr[107] = arr[107]+1 = 0 + 1 = 1



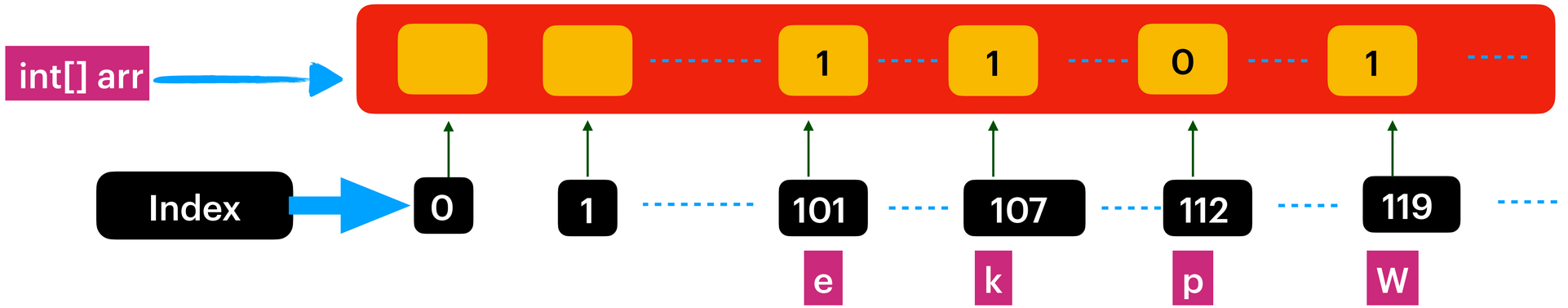

As arr[107(k)] is not > 1 so move rightIndex
max = Math.max(max, right-left+1)
max = Math.max(2,3-2+1) = 2

Solution is : SlidingWindow with
repeated move of leftPointer when there is duplicate character

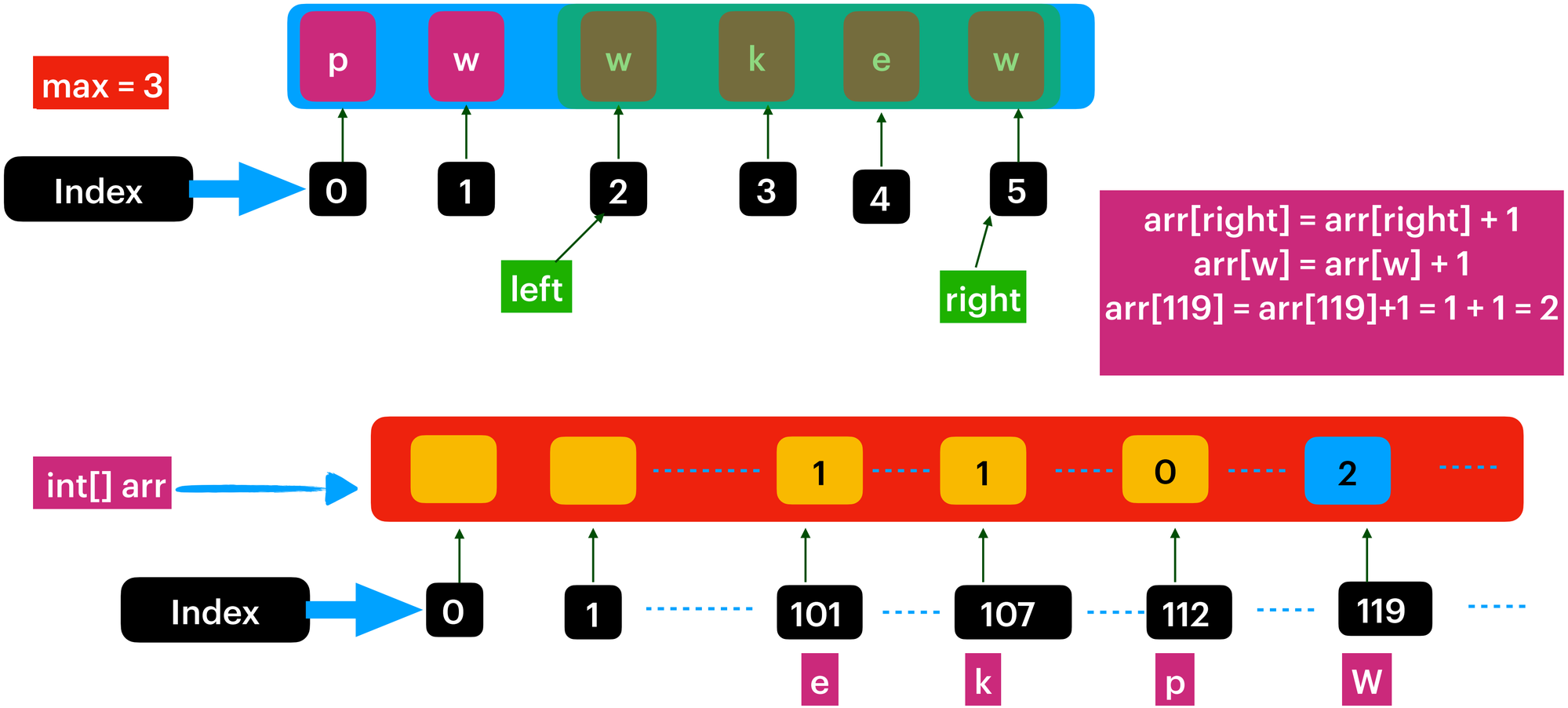
Take the int[] array of size 128

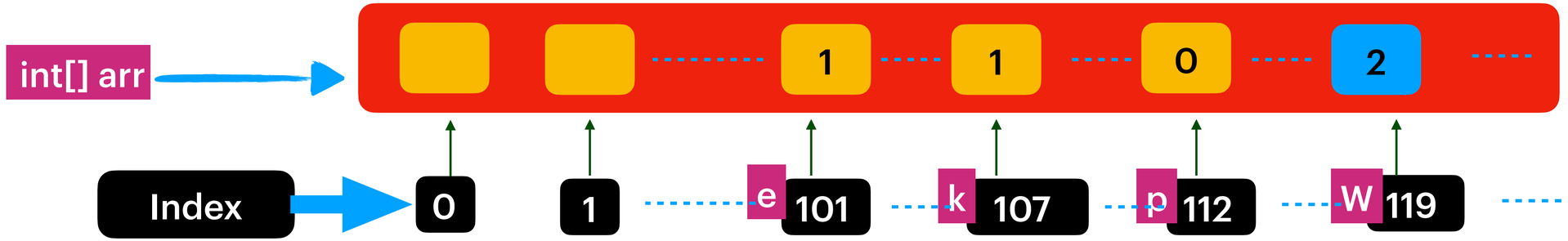


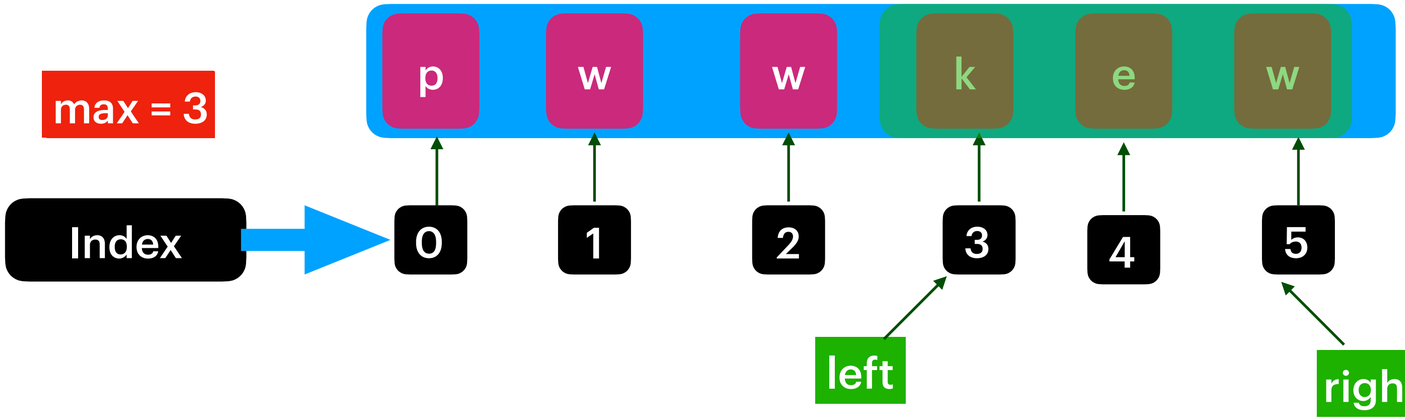

arr[right] = arr[right] + 1
arr[e] = arr[e] + 1
arr[101] = arr[101]+1 = 0 + 1 = 1




As arr[101(e)] is not > 1 so move rightIndex
max = Math.max(max, right-left+1)
max = Math.max(2,4-2+1) = 3

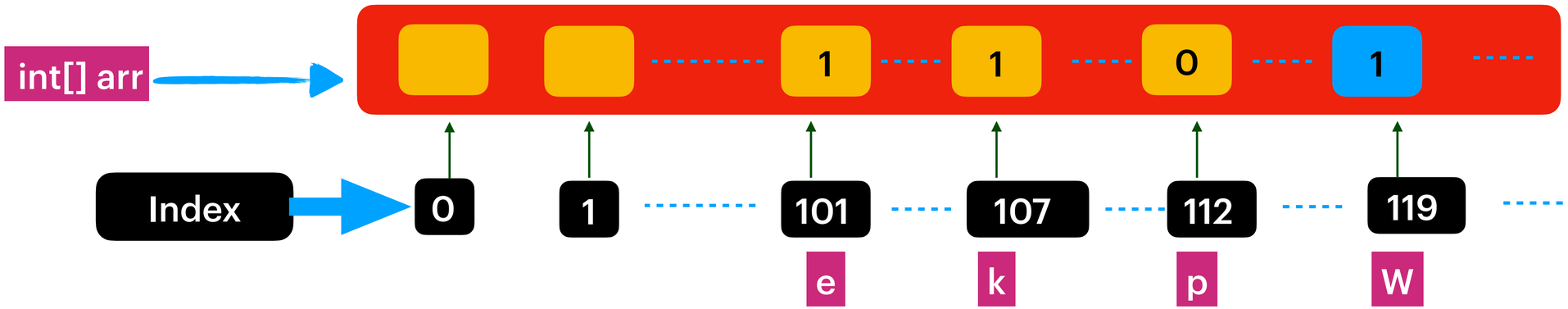

max = 3
p
w
w
k
e
w
Index
0
1
2
3
4
5
left
right
arr[right] = arr[right] + 1
arr[w] = arr[w] + 1
arr[119] = arr[119]+1 = 1 + 1 = 2
int[] arr
1
1
0
2
Index
0
1
101
107
112
119
e
k
p
W
As arr[119] is > 1 so move rightIndex :
Repeat below process till arr[119] <= 1
Decrement current right index value
move leftPointer to one
index forward .




As arr[(right)119] is > 1 so move rightIndex :

Repeat below process till arr[(right)119] <= 1
Decrement current left Pointer value
move leftPointer to one
index forward .




As arr[119] is not > 1 so move rightIndex
update the max = Math.max(max, right-left+1)
max = Math.max(3,5-3+1) = 3