Agenda :

1. Count distinct elements in subarray of size K
2. Pair with given sum
3. Largest consecutive sequence

↳ Doubts

dict = {}

dict[" '] = [ ]

{3 : 1
 2 : 4
 1 : 3}

len(dict)

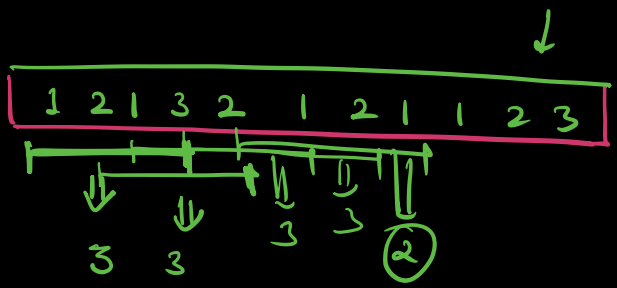
↓

no. of keys

Q Given N array elements. Calculate no. of distinct elements in every window of size K.

subarray

Input

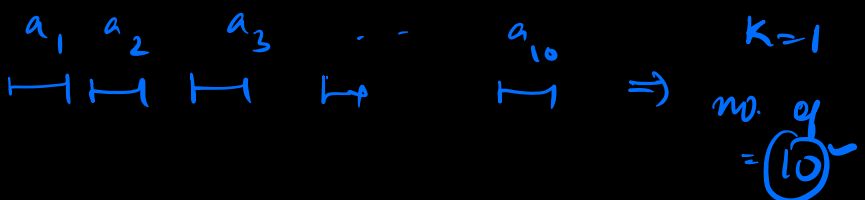


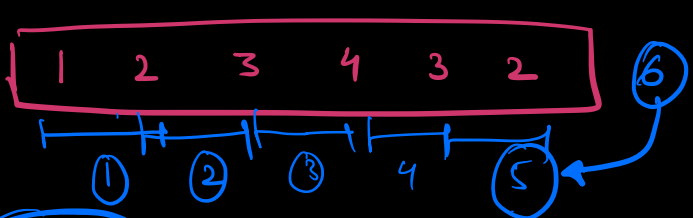
K = 4

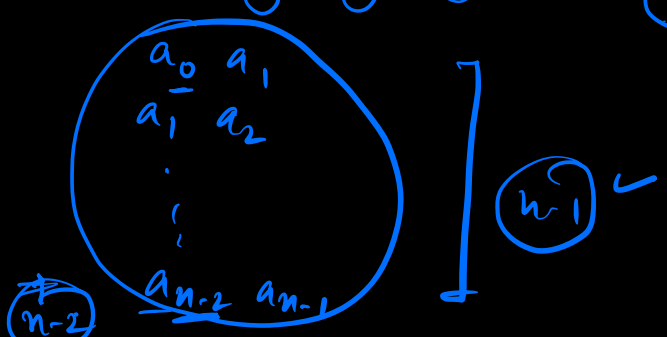
Example

return → ans = [3, 3, 3, 3, 2, 2, 2, 3]

N sized array → 10

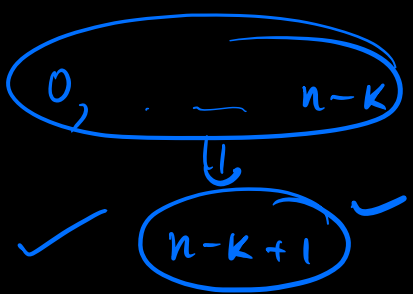


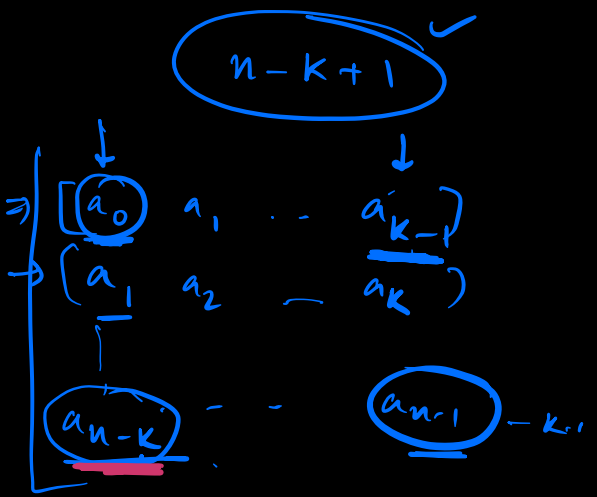




$k = 1 \Rightarrow n$

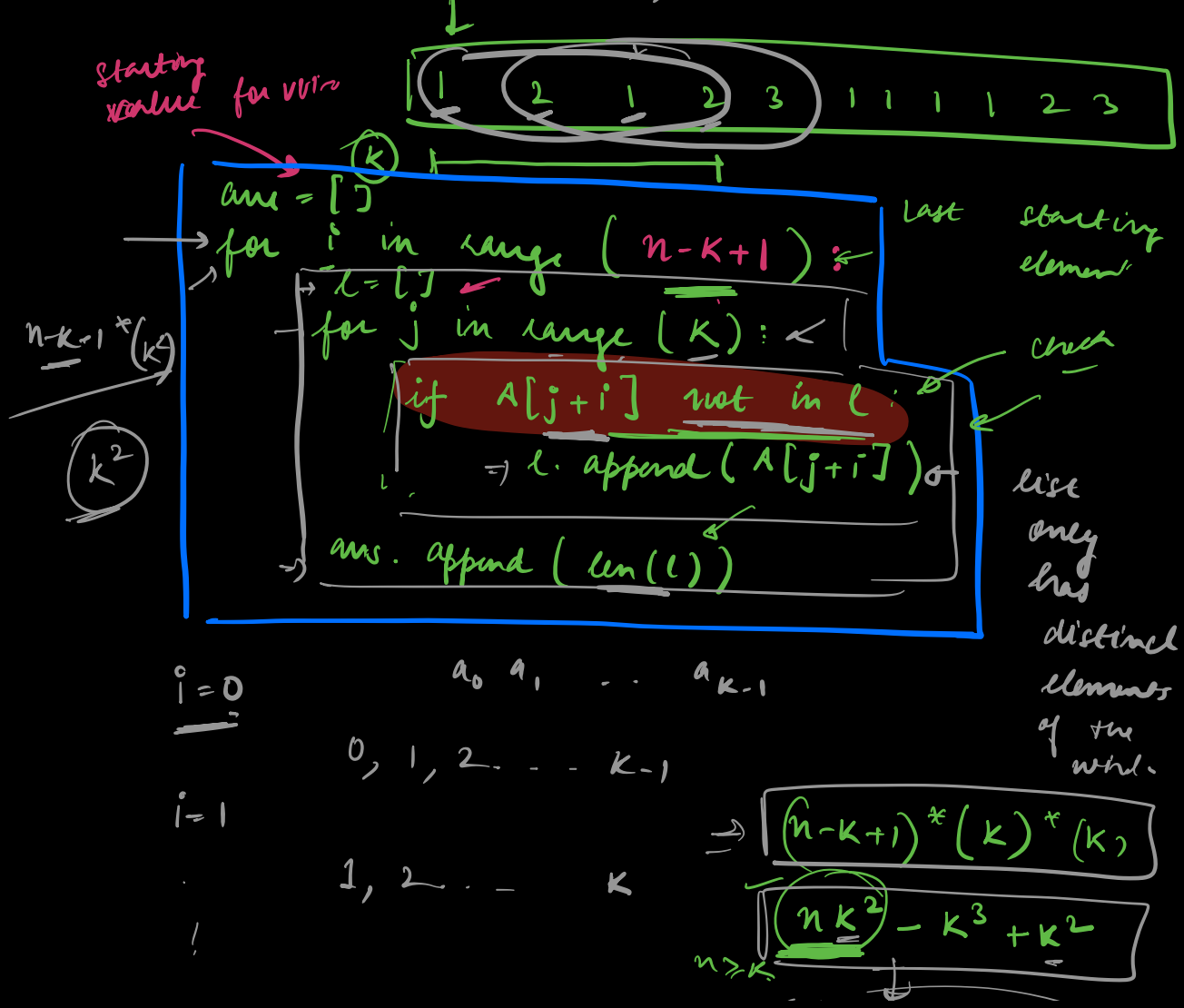
$k = 2 \Rightarrow n-1$





Brute force

$l = [1, 1, 3]$ ans = [2, 3



i = 0 $a_0\ a_1\ \ldots\ a_{k-1}$

0, 1, 2 . . . k-1

i = 1

1, 2 . . . k

.

.

!

$\Rightarrow (n-k+1) * (k) * (k)$

$nk^2 - k^3 + k^2$

$n \geq k$

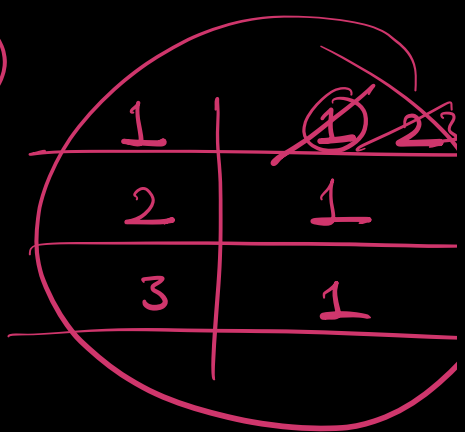
Optimisation - 1

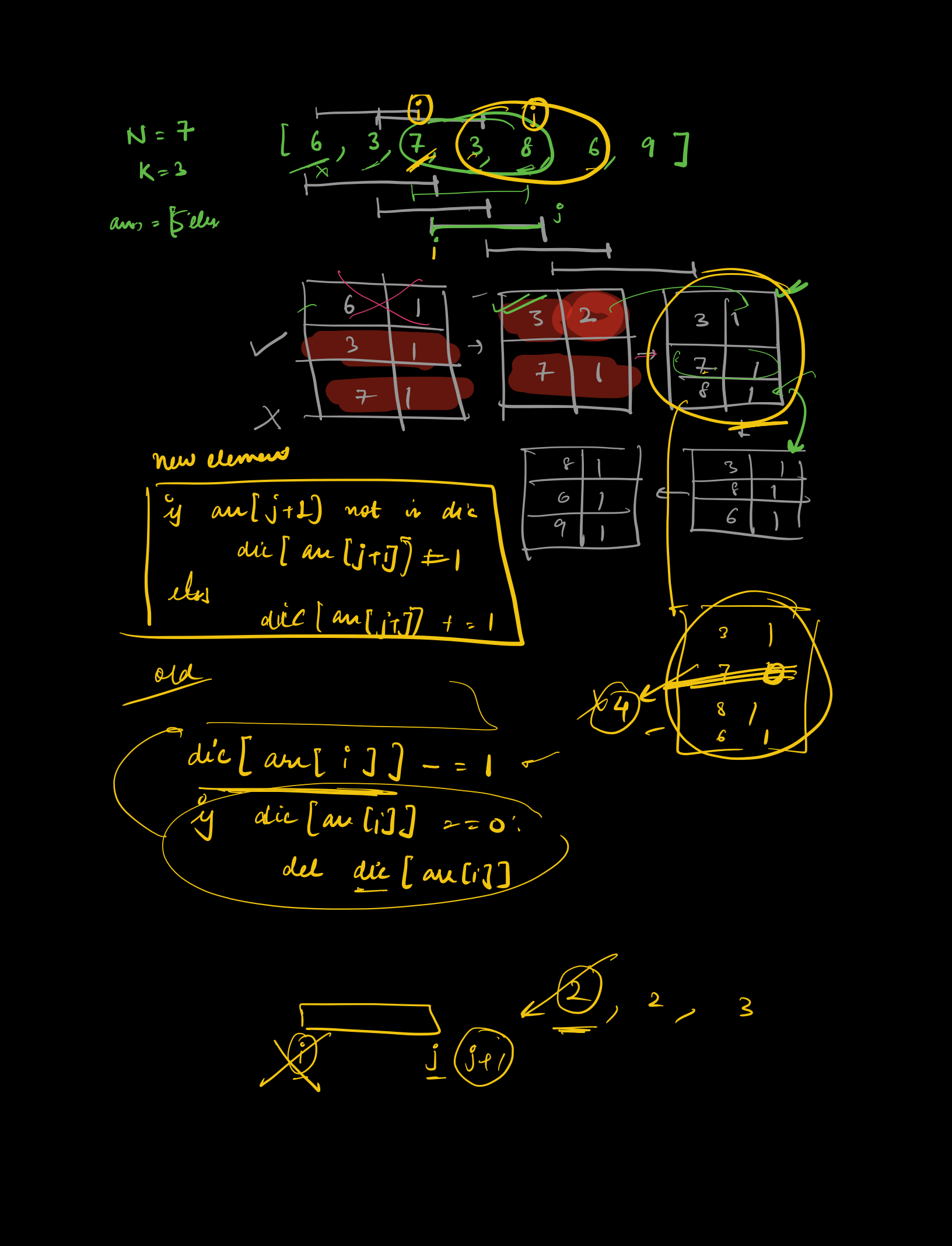
$O(nk^2)$ $O(N^3)$

```
for i in range (n-k+1):
    dic = {}
    for j in range (k):
        if arr[i+j] not in dic:
            dic [ arr[i+j] ] = 1
        else:
            dic [arr[i+j]] += 1
    ans.append ( len (dict))
```

can skip!

$(n-k+1) * k$ $= O(nk)$




N = 7
K = 3
arr = [6, 3, 7, 3, 8, 6, 9]
i
j
new element
if arr[j+1] not in dic
dic[arr[j+1]] += 1
else
dic[arr[j+1]] += 1
old
dic[arr[i]] -= 1
if dic[arr[i]] == 0:
del dic[arr[i]]
2, 2, 3
i
j
j+1

Optimised solution →

in collab

TC → O(N)

SC → O(K)

O(N-K+1) O(K)

Q Given N array elements. Check if there exist a pair (i, j) such that arr[i] + arr[j] == K

(i, j) → distinct

arr

[2, 7, 11, 13]

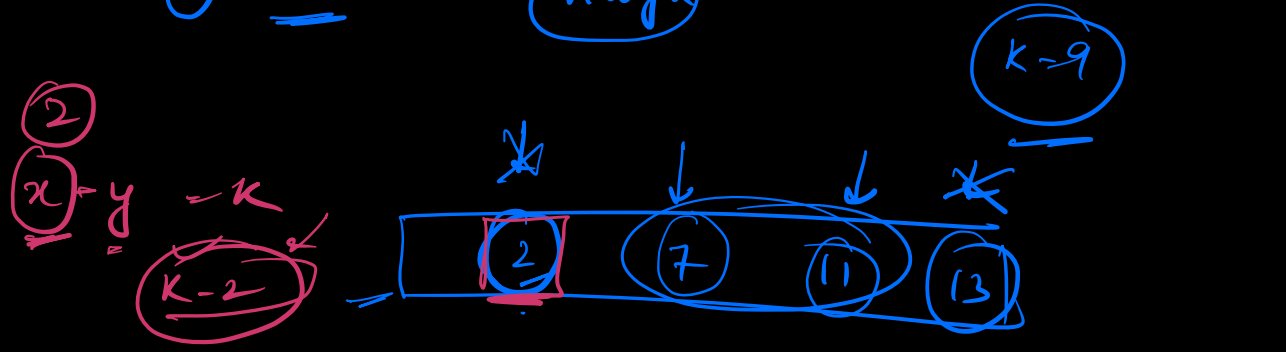
K = 18 → 7, 11

K = 9 → 2, 7

K = 17 ✗

1. BF iterate on all pairs

2. — nlogn



2 + 13 = 15 < 18

20 > 18

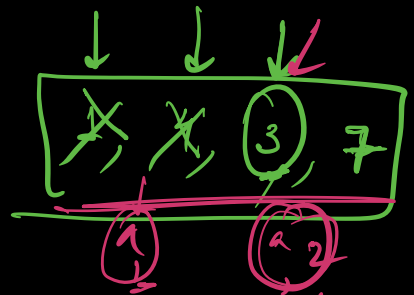
7, 11

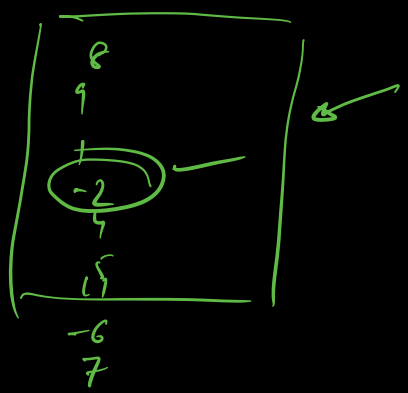
(3) Without sorting

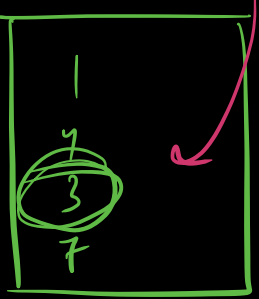
K = 6

[8, 9, 1, -2, 4, 5, 11, -6, 7, 5]

K = 6



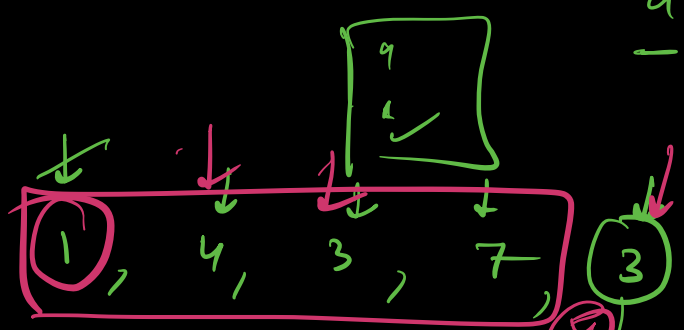




$a_1 + a_2 = K$

NO

a_1 a_2

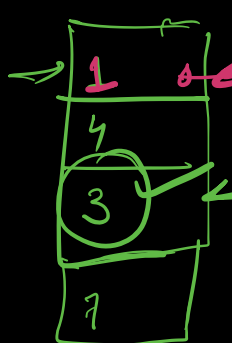


$x + y = K$

$y = K - 1$

Yes!

K - arr[i]



```
dic{}
for i in range(n)
    if k-arr[i] in dic:
        return True
    else:
        if arr[i] not in dic:
            dic[arr[i]] = 1
```

~~6+2 =~~

k = -10

-9

-9 + (x)

= -10

x = -1

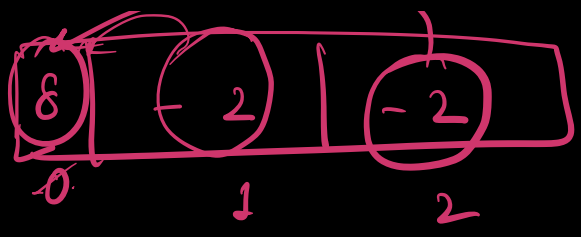
k - arr[i] is same as arr[i]

8, -2, -2 k = 6

H.W

Count the no. of pairs so

a[i] + a[j] = = K



(8, -2)

1 pair

1 pair

2 pa.

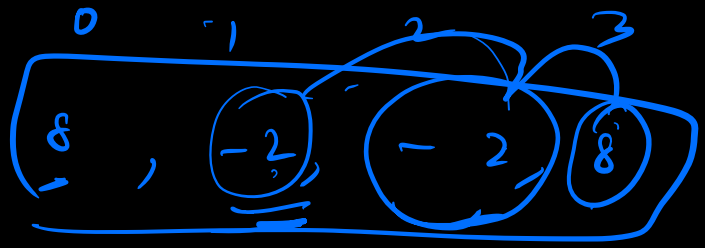
8 → 1

-2 → 2

(0, 1) (1, 0)

(0, 2) (2, 0)

pairs (i, j)



8 → 2

-2 → 2

8 ~~→ 2~~ (0, 1)

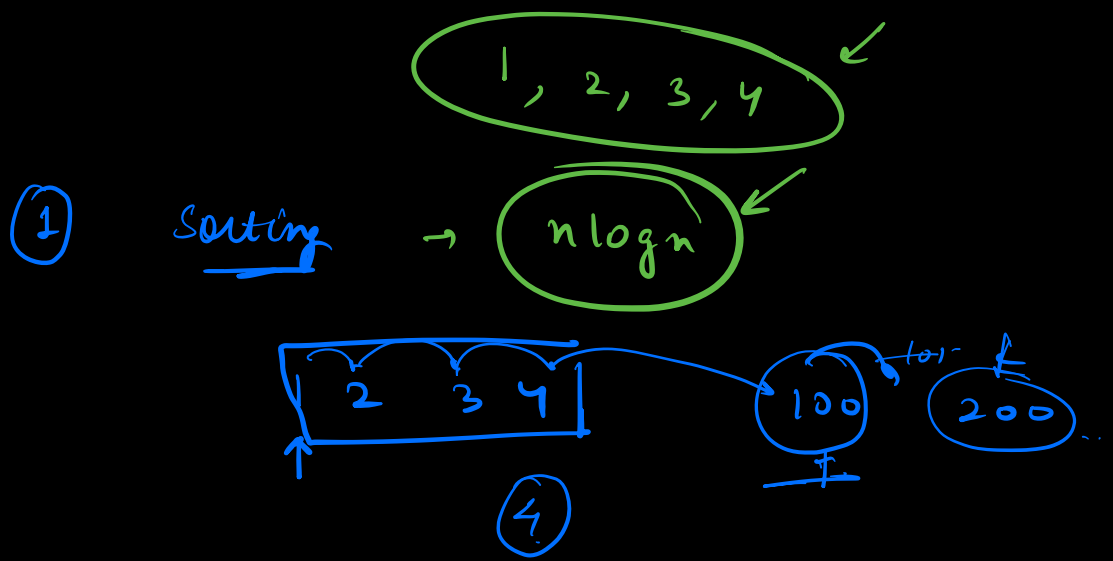
(0, 2)

(3, 1)

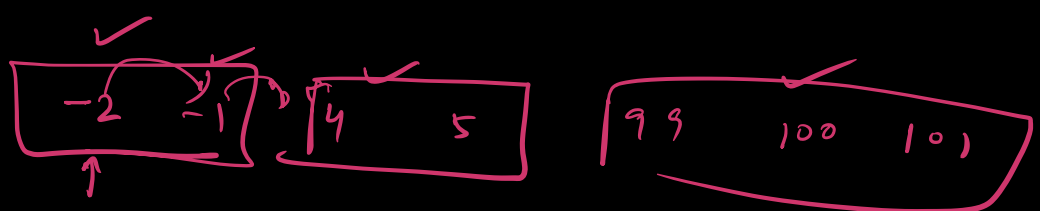
(3, 2)

Q Given N elements. Find the length of longest sequence which can be rearranged to form a sequence of consecutive ~~nos~~ .. increasing no.s.

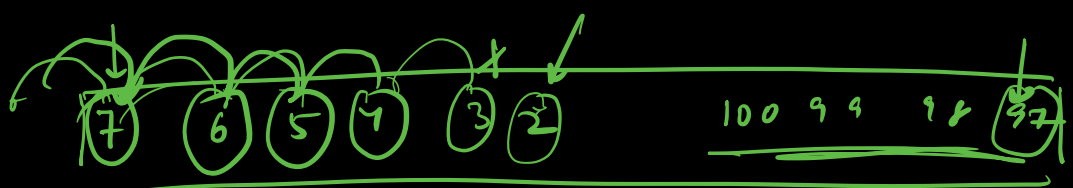
[100, 4, 200, 1, 3, 2]

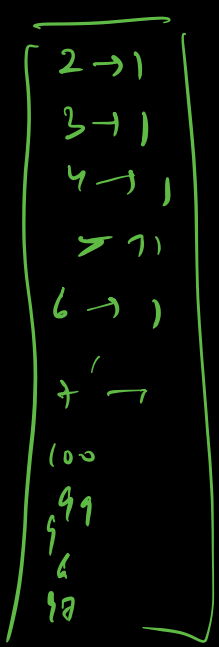


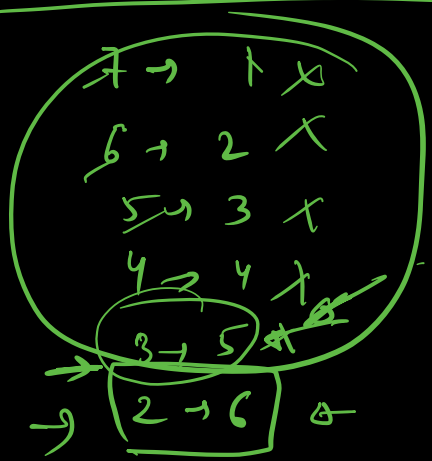
-2, -1, 5, 4, 101, 99, 100

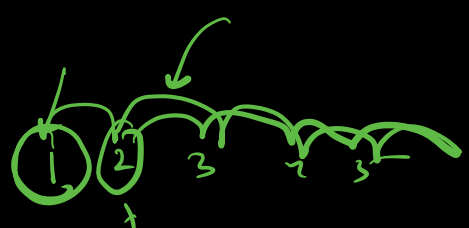


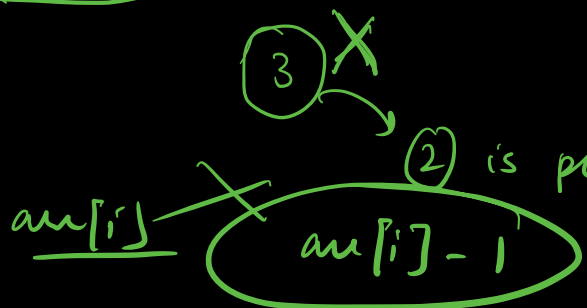
```
arr.sort()
i = 0
while i < N :
    cnt = 1
    prev = arr[i]
    i += 1
    while arr[i] == prev + 1 :
        prev += 1
        i += 1
        cnt += 1
    ans = max(ans, cnt)
```











① create dictionary of elements

② for i in range(n): ↙

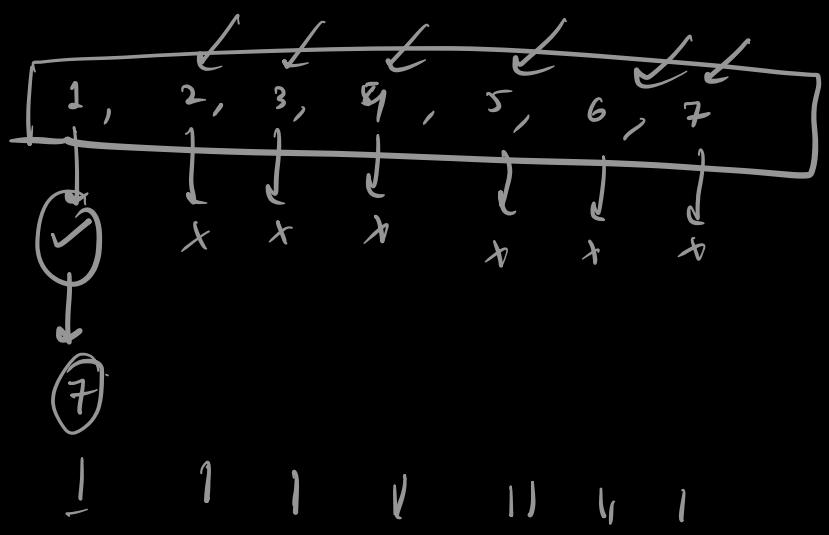
if arr[i] - 1 in dict:

continue:

else:

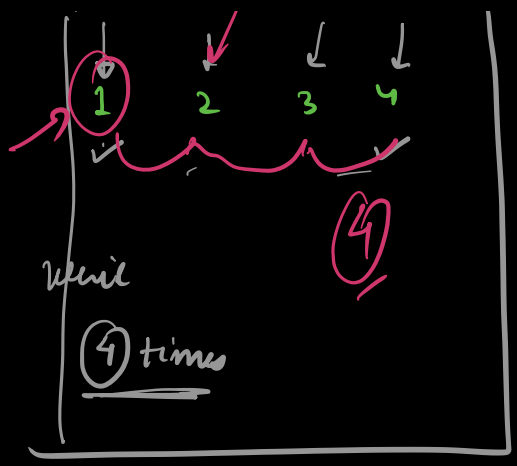
```
val = arr[i]
cnt = 0
while val in dict:
    cnt += 1
    val += 1
ans = max(ans, cnt)
```

2 → 3 → 4 → 5 →

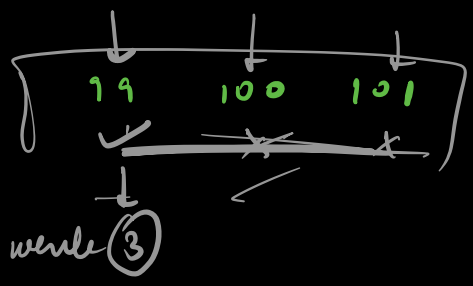


= 2N iterations

O(N)

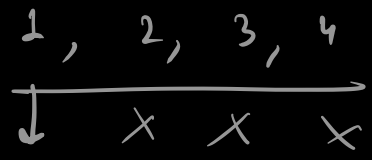


8 iterations

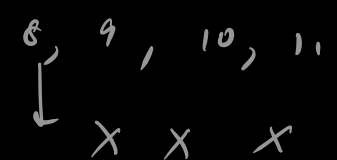


= 6 iterations

αN iterations

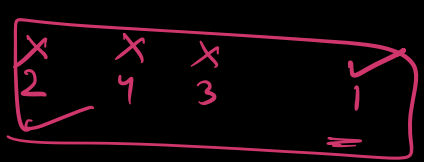


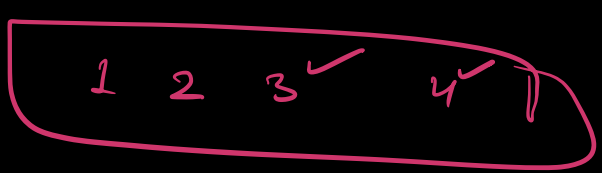
4

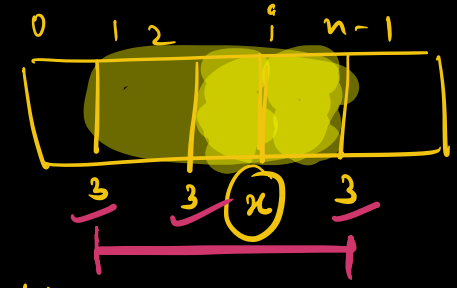


4

= 8 times







die :) all the previous prefix sum

k - prefix sum

v (?)