Lists2 batch14

Q. Given an arr, rotate from right to left k-times.

I/P = 3 -2 1 4 6 9 8 k = 3

k=1 8 3 -2 1 4 6 9

k=2 9 8 3 -2 1 4 6

k=3 6 9 8 3 -2 1 4 = O/P

Ex → I/P 4 1 6 9 2 14 7 8 3 k = 4

14 7 8 3 4 1 6 9 2

Ex I/P -2 3 1 4 6 2 8 7 9 3 k = 3

O/P 7 9 3 -2 3 1 4 6 2 8

I/P = 3 -2 1 4 6 9 8 k = 3

k=1 3 3 -2 1 4 6 9 temp = 8

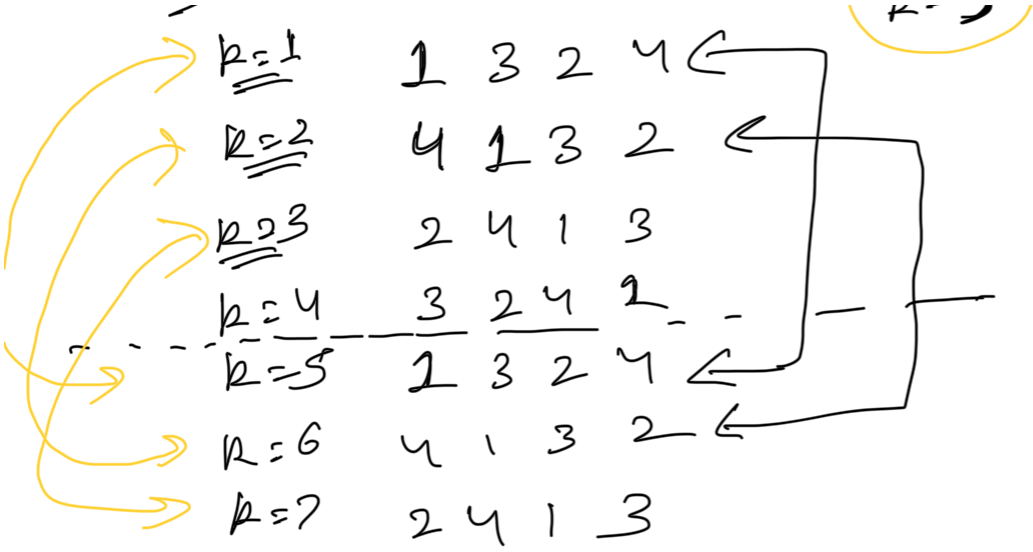
8

k=2

k=3

I/P = 3 2 4 1 length = 4

k = 5



$k = k \% \text{length(arr)}$

$= 7 \% 4 = 3$

1 rotation $\rightarrow O(N)$

k rotation $\rightarrow O(kN)$

-2 3 1 4 6 2 8 7 9 3 $k=3$

1) Reverse entire array = 3 9 7 8 2 6 4 1 3 -2
2) Reverse 1st k de = 7 9 3 8 2 6 4 1 3 -2
3) Reverse last N-k de = 7 9 3 -2 3 1 4 6 2 8

entire arr

$T.C. = O(N + k + N-k) = O(2N) = O(N)$

$S.C. = O(1)$

Reverse array

Q Given an arr & an integer k, return total no. of subarrays whose sum equals k.

Ex nums = [1, 2, 3], k = 3

Subarrays
- [1] → 1 ✗
- [2] → 2 ✗
- [3] → 3

O/P = 2

[1,2] → 3
[2,3] → 5 ✗
[1,2,3] → 6 ✗

⇨ [2 5 1 6 8]

Subarrays

[5 1] ✓
[5 1 6 8] ✓
[5 6 8] ✗
[5 8] ✗
[6] ✓
[6 8] ✓
[2 5 1] ✓
[2 1] ✗
[1 6 8] ✓
[2 5 1 6 8] ✓
[8 6 5] ✗
[5 1 6] ✓
[5 6] ✗

Subsequences

[2 6] ✓
[5 6 8] ✓
[2 1 6 8] ✓
[2 5 8 6 1] ✗
[1 8] ✓
[2 5 1 6] ✓
[6 1 5 2] ✗

Subsets

[5 2 1] ✓
[5 6] ✓
[5 5] ✗
[1 8 2 5 6] ✓
[1 2 8] ✓

* (5 2 1) = (1 2 5)
= (2 1 5)

(5 5 1 2) ⇒ (5 1 2)

(5 1 2) → (1 2 5)
(2 8 1)

(5 5 1 2)
(5 1 2)

Ex nums = [1, 2, 3], k = 3

Subarrays

[1] → 1 ✗
[2] → 2 ✗
[3] → 3
[1,2] → 3
[2,3] → 5 ✗
[1,2,3] → 6 ✗

O/P = 2

Ex

0	1	2	3	4	5	6	7	8	9
1	4	1	2	2	-1	4	-1	3	4

k = ?

[4,1,2] = 7
[3,4] = 7
[2 2 -1 4] = 7
[2 2 -1 4 -1 3 4] = 7

pf-sum

0	1	2	3	4	5	6	7	8	9
1	5	6	8	10	9	13	6	9	13

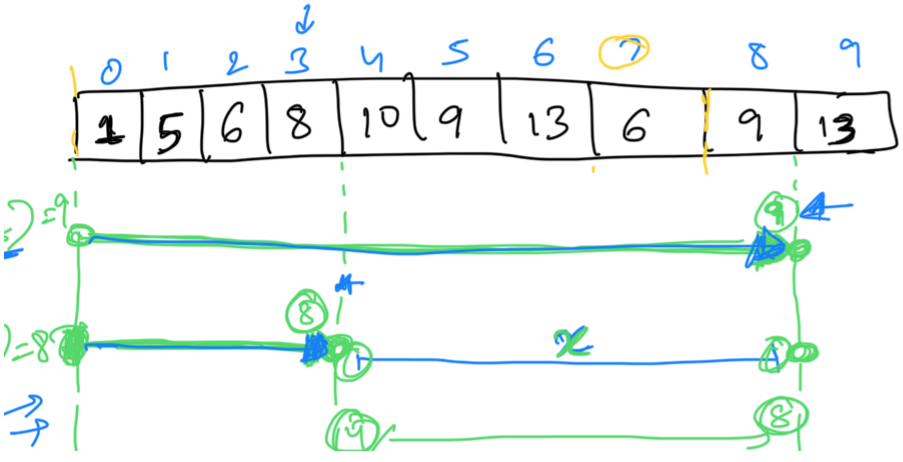
pf.sum[i] = Sum all the elements starty from idx 0 to i in original array.

pf-sum[7] = 6 = Sum of arr[0:7] (inclusive) is 6

pfsum[5] = 9 = Sum of all elements idx 0 to 5 = 9

Sum of all endants from any idx i to j ?

Sum of all elemets from 6 to 7 =



$8+x=9$

$x=9-8=1$

$8+x=9$

$x=9-8=1$

We are able to find sum of elements from 4 to 8

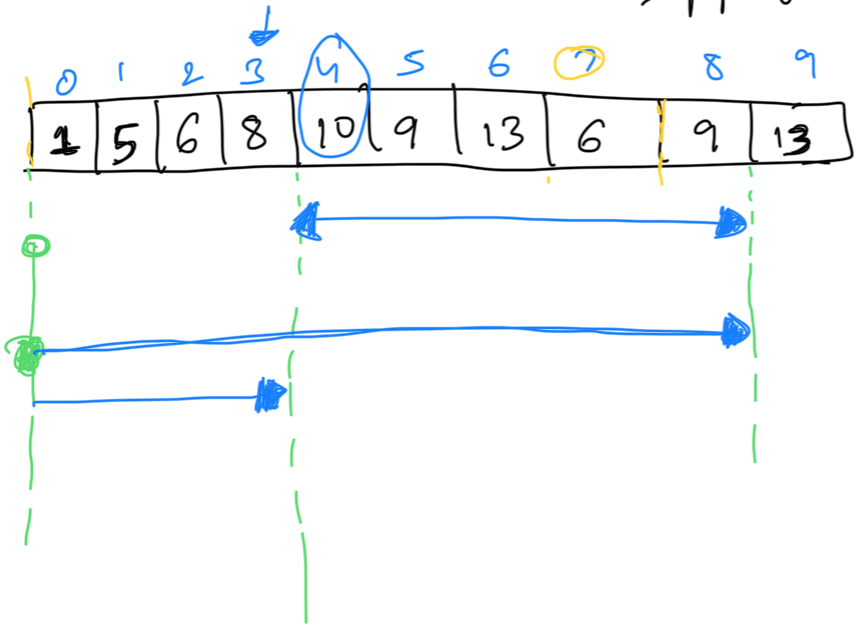
sum of elements i to j

pf sum --- 0 to j

i to j ⇒ ?

Sum of all elements from any idx i to j ?

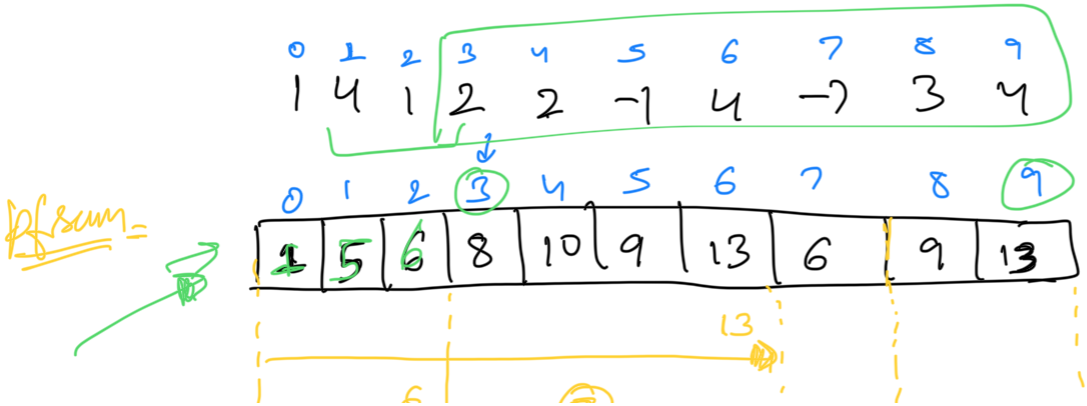
↳ pf[j] - pf[i-1]

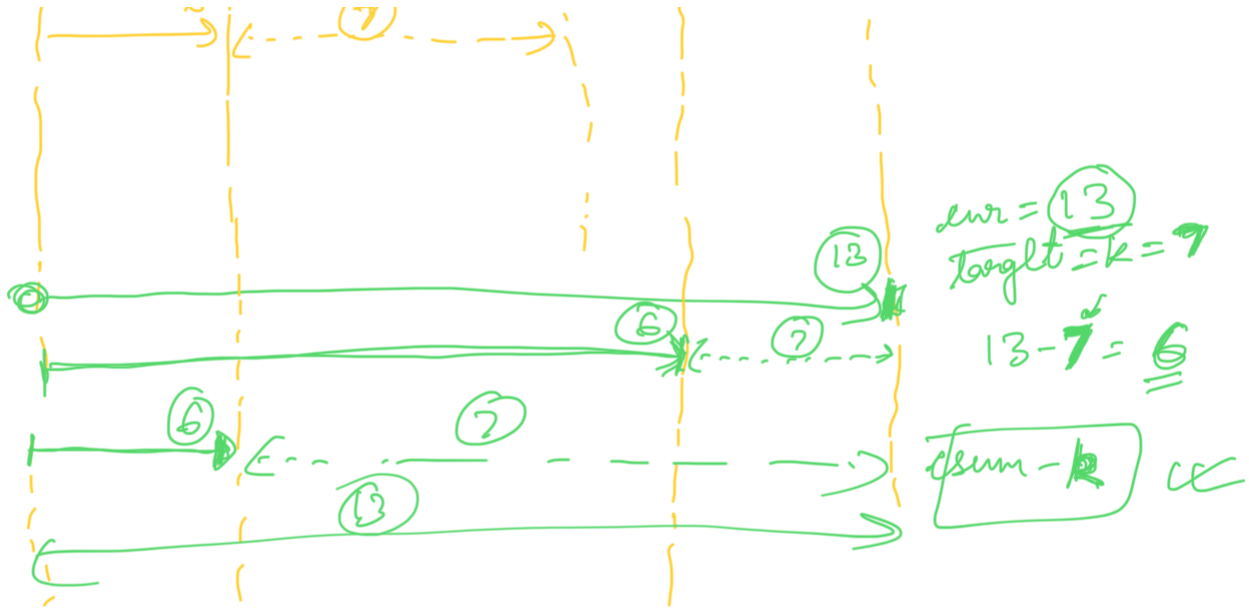


Sum of all the elems. 4 to 8

pf[8] - pf[3]

pf[j] - pf[i-1]

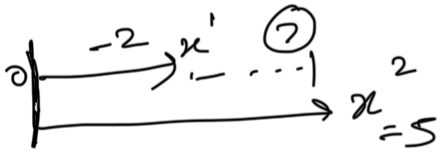


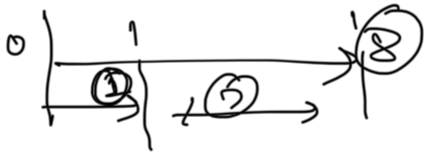


memory

set = {1, 5, 6, 8, 10, 9, 13, 6, 9, 13}

8 - 7 = 1 ; 13 - 7 = 6 ; -1 ; 2 ✗ ; 6 ✓





k = 7

0	1	2	3	4	5	6	7	8	9
1	4	1	2	2	-1	4	-7	3	4

ans 1 2 3

pfsum = 0

pfsum = 1 ; 1 - 7 = -6 ✗

pfsum = 4 + 1 = 5 ; 5 - 7 = -2 ✗

pfsum = 6 ; 6 - 7 = -1 ✗

pfsum 8 ; 8 - 7 = 1 ✓

10 ; 10 - 7 = 3 ✗

9 ; 9 - 7 = 2 ✗

13 ; 13 - 7 = 6 ✓

17 ; 17 - 7 = 10 ✓

hm

'0' : 1

'1' : 1

'5' : 1

'6' : 1

'10' : 1

'9' : 1

'13' : 1

'17' : 1

3

```
def subs_sum(nums, k):
    hm = {
            0 : 1
        }
    ans = 0
    pfsum = 0
    for num in nums:
        pfsum += num
        lookup = pfsum - k
        if lookup in hm:
            ans += hm[lookup]
        if pfsum not in hm:
            hm[pfsum] = 0
        hm[pfsum] += 1
    return ans
```

TC → O(N)
SC → O(N)

X——X

[2 6] k = 7

pf = 2 2-7 = -5X

pf = 8 8-7 = 1

{'2': 1}

lookup

3