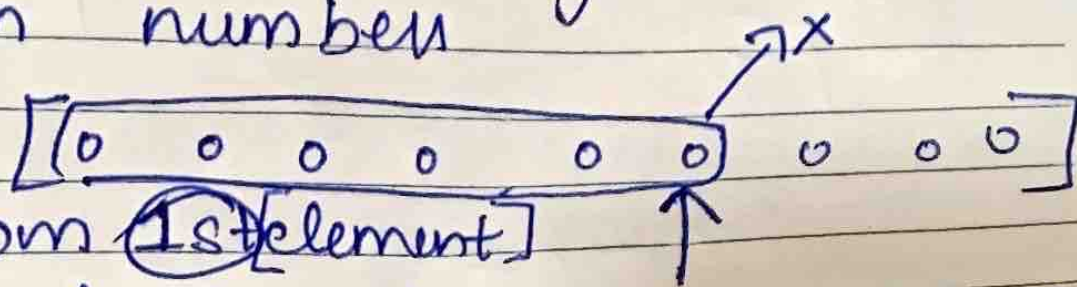


# Flashing METHOD

[ 0 0 0 0 0 0 0 ]

Array with random no.s.  
sum = k

Basically take an array with  
random numbers



Imagine from 1st element  
do the arrowed element,  
the sum is x taking  
into account that  
in that array

take this element  
there are ~~6~~ 6 elements  
6 elements

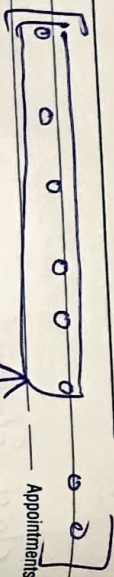
March

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					



February  
2015

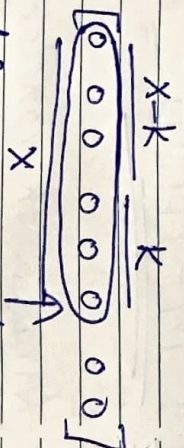
10  
Tuesday



Examining that array again.

The sum of the total elements is  $x$ , the required sum is '3', so the remaining sum

would be  $(x-k)$



Using this  $x-k$  (also known as prefix sum) we would be able to determine

if the forward array has a sum of  $k$ .

February

M	T	W	T	F	S	S	M	T	W	T	F	S	S
							1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28							

11  
Wednesday

for Example

Appointments

array = [1, 2, 3, 1, 1, 1, 1, 4, 2, 3]

$k=3$

sum = 0 (initially)

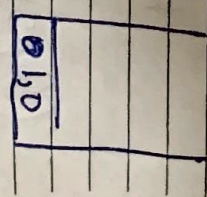
for each long = 0. sum = sum + arr[i]

Step 1

sum = 0 + 1 @ index 0

sum = 3

Not app

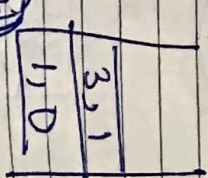


Step 2.

sum = 3 @ index 1

arr [long = 2] if (sum == k)

long = math.max (long, i+1)



March

M	T	W	T	F	S	S	M	T	W	T	F	S	S
							1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31				

if arr[i+1] since initially from @.



February 2015

R

L

043-322

Thursday

12

WK - 7

Step 3

sum = 6 @ index = 3

sum - k = 3

6 - 3 = 3

contains in hash map

But

(2 > 1) ∴ long = 2

6, 2
3, 1
1, 0

Appointments

Step 4

sum = 7

sum - k = 4 which doesn't contain in hash

kernel just update the hashmap

7, 3
6, 2
3, 1
1, 0

Step 5

sum = 8

sum - k = 5 which doesn't contain in (hash map)

8, 4
6, 2
3, 1
1, 0

kernel just update in map

February

MTWTFSSMTWTFSS  
1 2 3 4 5 6 7 8  
9 10 11 12 13 14 15 16 17 18 19 20 21 22  
23 24 25 26 27 28

Friday

13

J

044-321

February 2015

Appointments

Step 6

sum = 9

sum - k = 6

long = 2 till now

long = max(2, i - map.get(sum - k))

8, 4
7, 3
6, 2
3, 1
1, 0

exists in map

And so on

Here the code would be

fun(i) (arr, long k)

long sum = 0;

long long = 0;

for (int i = 0; i < n; i++)

{ sum = sum + arr[i];

if (sum == k)

{ ~~long~~ long = map.get(sum - k)

if (map.containsKey(sum - k))

{ long = max(long, i - map.get(sum - k))

March

MTWTFSSMTWTFSS

1 2 3 4 5 6 7 8  
9 10 11 12 13 14 15 16 17 18 19 20 21 22  
23 24 25 26 27 28

map.put(sum, i);