2nd January 2022

Today's Task

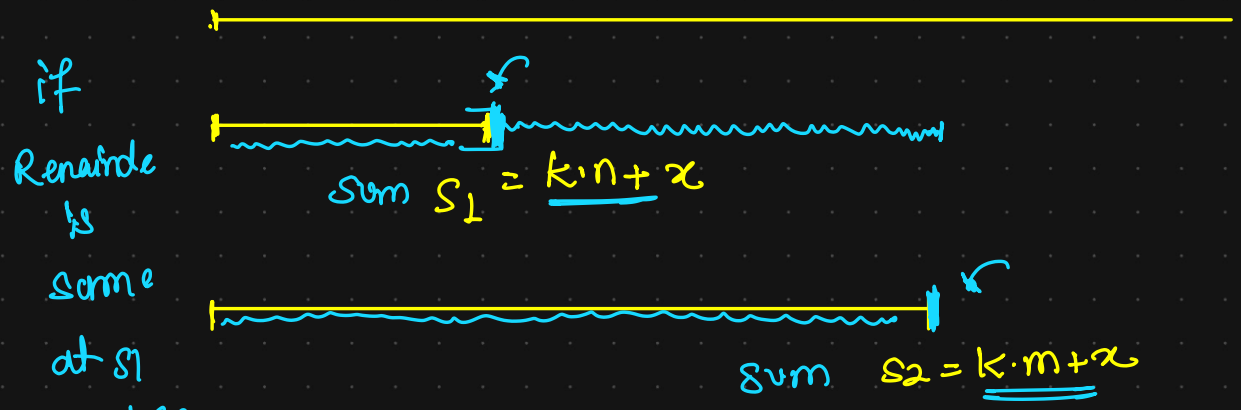
1. Longest subarray with sum divisible by k
2. count of subarrays with sum divisibly by k
3. Longest subarray with equal number of 0s, 1s and 2s.
4. count of subarrays with equal number of 0s, 1s and 2s.
5. Task completion

Longest Subarray with sum divisible by k : →

array → 2 7 6 1 4 5

k = 3

length = ?



$S_1 = k \cdot n + x$ (x: Remainder)

$S_2 = k \cdot m + x$ (x: Remainder)

k = 5

Q3

5 × 16 + 3

k * n + x

Sum of middle part $= S_2 - S_1$

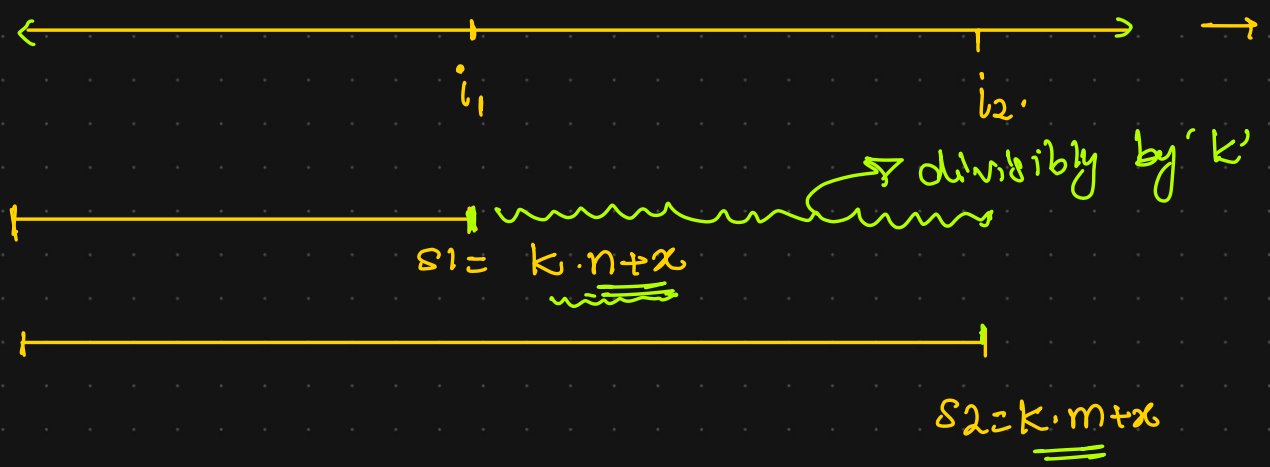
$= k \cdot m + x - (k \cdot n + x)$

$= k \cdot m + x - k \cdot n - x$

$= k(m - n)$

NOTE!

⇒ Sum of middle part is divisible by k



Sum → HashMap → Remainder

→ Remainder is same for both cyclic sum, s_1 & s_2 proves

Key

To prove: sum between i_1 to i_2 is divisible by k → subarray from i_1 to i_2.

$i_1 < i_2$.

$s_1 = k \cdot n + x$ & s_1 is sum of elemt from 0 to i_1

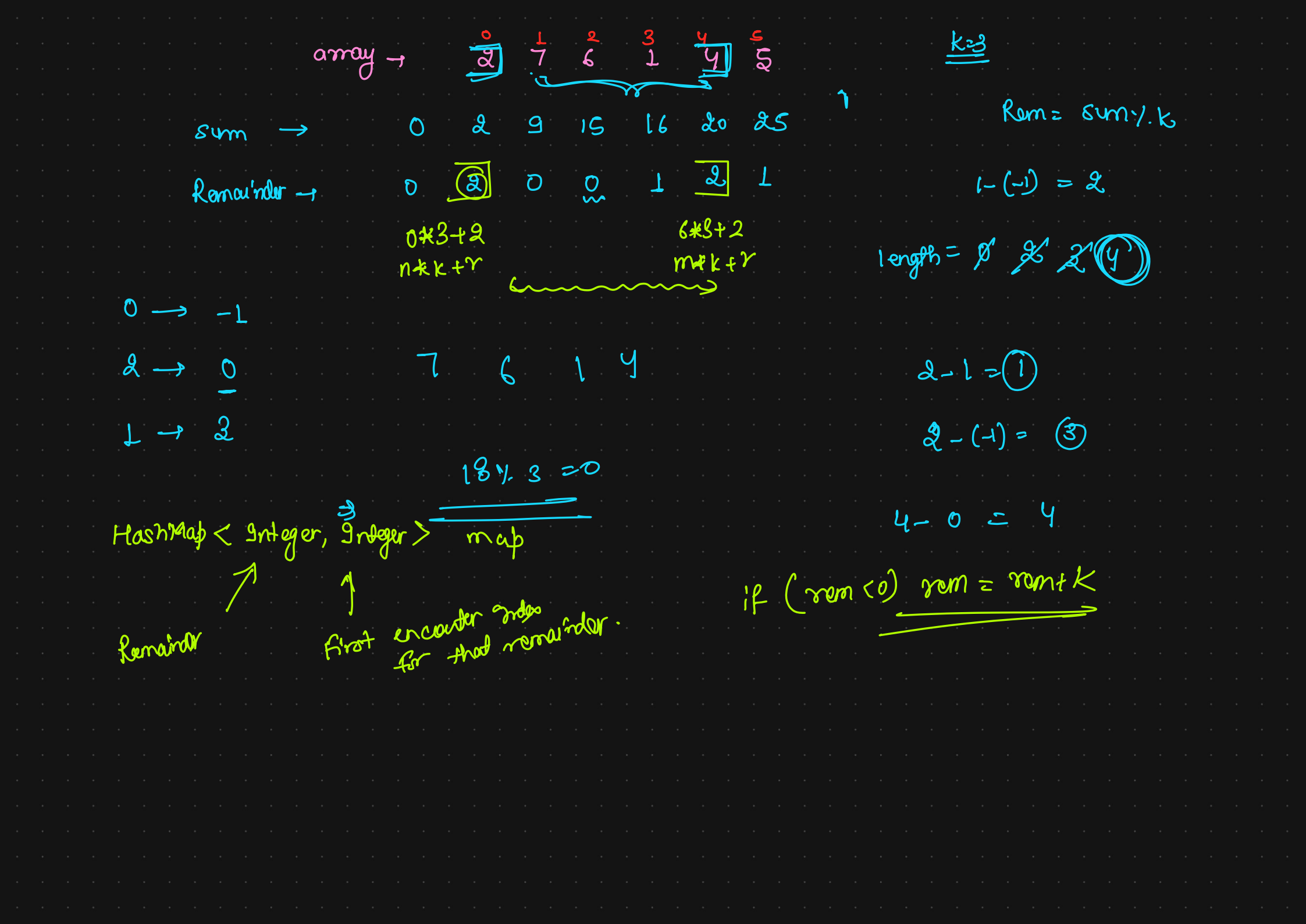
$s_2 = k \cdot m + x$ & s_2 is sum of elemt from 0 to i_2.

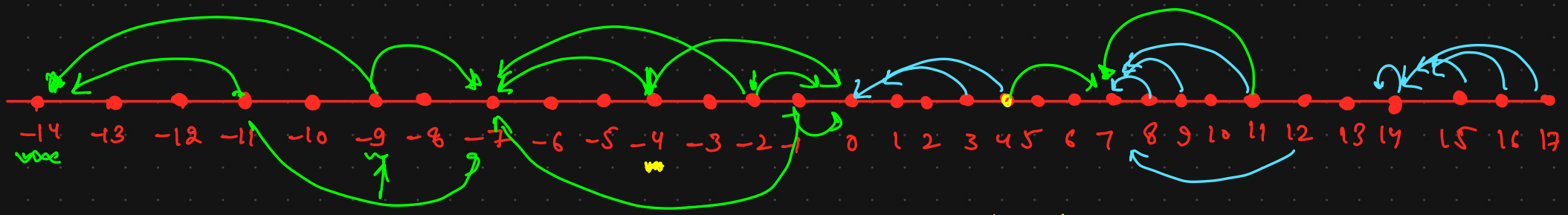
s_3 (sum from i_1 to i_2) $= s_2 - s_1 = (k \cdot m + x) - (k \cdot n + x)$

$= k \cdot m + x - k \cdot n - x$

$s_3 = k(m - n)$

$\Rightarrow$ s_3 is divisibly by k.


array → 2 7 6 1 4 5
0 1 2 3 4 5
k=3
sum → 0 2 9 15 16 20 25
Remainder → 0 2 0 0 1 2 1
Rem = sum%k
1-(-1) = 2
0*3+2
n*k+r
6*3+2
m*k+r
length = 2 3 4
0 → -1
2 → 0
1 → 2
7 6 1 4
2-1 = 1
2-(-1) = 3
18%3 = 0
HashMap<Integer, Integer> map
4-0 = 4
Remainder
first encounter index for that remainder.
if (rem<0) rem = rem+k



$-9 \,\%\, 7 = -2$

$k = 7$

$11 \,\%\, 7 = 4$

$4 \,\%\, 7 = 4 - 7 = -3$

remainder

generic behaviour

$-4 \,\%\, 7 = -4$ OR $+3$

$-2 \,\%\, 7 = -2$ OR 5

$6 \,\%\, 7 = 1$

$9 \,\%\, 7 = 2$

$3 \,\%\, 7 = 3$

$4 \,\%\, 7 = 4$

$-9 \,\%\, 7 = -2$ OR $+5$

$-11 \,\%\, 7 = -4$ OR (3)

$-1 \,\%\, 7 = -1$ OR (6)

$11 \,\%\, 7 = 4$

$12 \,\%\, 7 = 5$

$14 \,\%\, 7 = 0$

$15 \,\%\, 7 = 1$

$16 \,\%\, 7 = 2$

$17 \,\%\, 7 = 3$

$-4 \,\%\, 7 = -3$

$|val| \,\%\, 7 =$ ... (rem)

(sign of a)

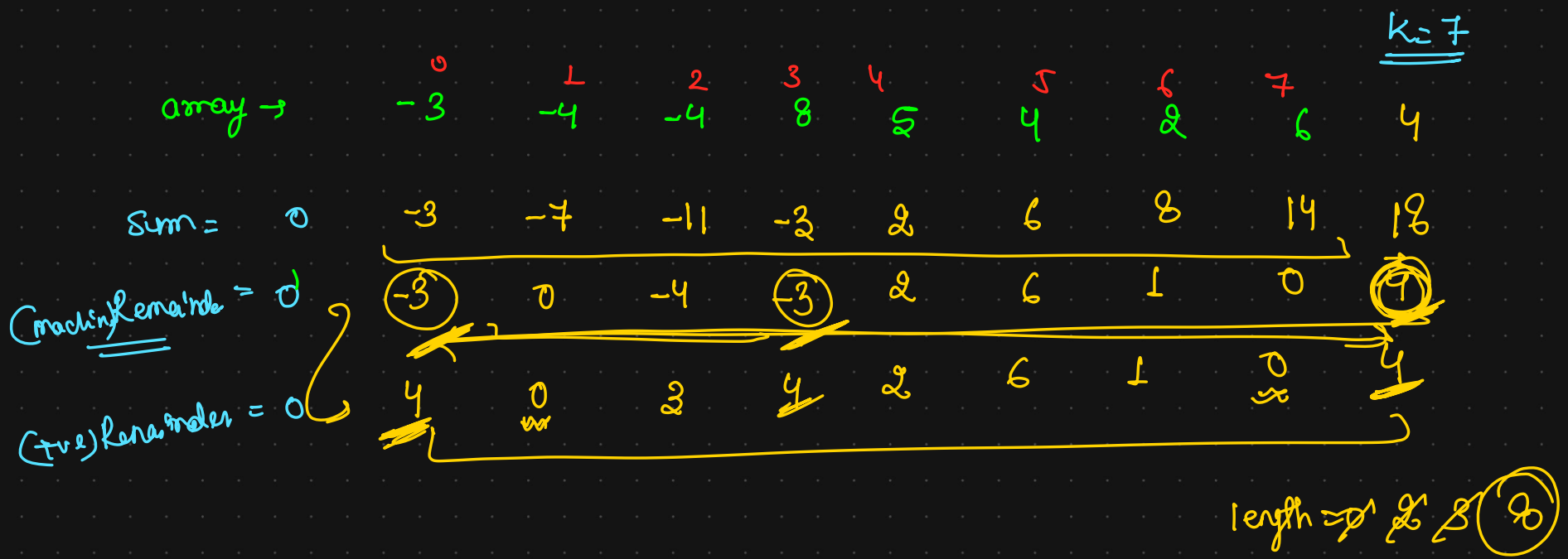
$-4 \,\%\, 7 = 4 \,\%\, 7 = -3$

$-9 \,\%\, 7 = -2$

generalise -

⇒ if remainder -ve

↳ +ve → add k



0 → -1

4 → 0

3 → 2

2 → 4

6 → 5

1 → 6

-ve Remainder -

$-k < rem < 0$

add k both

+ve → $-k + k < rem + k \leq k$

$0 \leq rem2 < k$

Count of subarrays with sum divisible by K:→

k=5.

array →	2	4	8	1	7	3	6	1	9	2	7	3
sum →	0	2	6	14	15	22	25	31	32	41	43	50	53
Remainder →	0	2	1	4	0	2	0	1	2	1	3	0	3

Rem → Freq.

0 → ~~1~~ ~~2~~ ~~3~~ 4

2 → ~~1~~ ~~2~~ 3

1 → ~~1~~ ~~2~~ 3

4 → 1

3 → ~~1~~ 2

Count = ~~0~~ ~~1~~ ~~2~~ ~~4~~ ~~5~~ ~~7~~ ~~9~~ ~~12~~ 13

→ 2 4 8 1

→ 4 8 1 7

→ 7 3

→ 2 4 8 1 7 3

→ 8 1 7 3 6

→ 3 6 1

→ 4 8 1 7 3 6 1

→ 1 9

→ 8 1 7 3 6 1 9

→ 6 1 9 2 7

→ 7 3 6 1 9 2 7

→ 2 4 8 1 7 3 6 1 9 2 7

→ 7 3

	0	1	2	3	4	5	6	7
array →	-3	-4	-4	8	5	4	2	6
sum →	0 -3	-7	-11	-3	2	6	8	14
rem → (-ve)	0 -3	0	-4	-3	2	6	1	0
rem → (+ve)	0 4	0	2	4	2	6	1	0

k=7

count = 0 X 2 4

0 → X 2

4 → X 2

3 → 1

2 → 1

6 → 1

1 → 1

→ -3 -4

→ -4 -4 8

→ -4 8 5 4 2 6

→ -3 -4 -4 8 5 4 2 6

Longest Subarray with equal number of 0s, 1s and 2s →

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
array →		0	0	1	2	2	1	0	0	1	1	2	1	2	2	0	2	2	1	1	1
0's →	0	1	2	2	2	2	2	3	4	4	4	4	4	4	4	5	5	5	5	5	5
1's →	0	0	0	1	1	1	2	2	2	3	4	4	5	5	5	5	5	5	6	7	8
2's →	0	0	0	0	1	2	2	2	2	2	2	3	3	4	5	5	6	7	7	7	7
key	0#0	-1#0	-2#0	-1#-1	-1#0	-1#1	0#0	-1#0	-2#0	-1#-1	0#-2	0#-1	1#-2	1#-1	1#0	0#0	0#1	0#2	1#1	2#0	3#-1

equal no. of 0, 1 & 2

→ Separator

0's > 1s → 2 more

2 & 1 → equal

-2#0

count of 1s – count of 0s

Key → a#b → a#b ↘ count of 2s – count of 1s

↳ string. ↳ separator

C1s - C0s

C2s - C1s

value → index

longest

What? Why? How?

Count of Subarrays with equal no. of 0s, 1s and 2s ⟶

Count is same as previous all problems,

we manage count 0, count 1 and count 2.

Make key in format of a # b.

a ⟶ count 1 − count 0

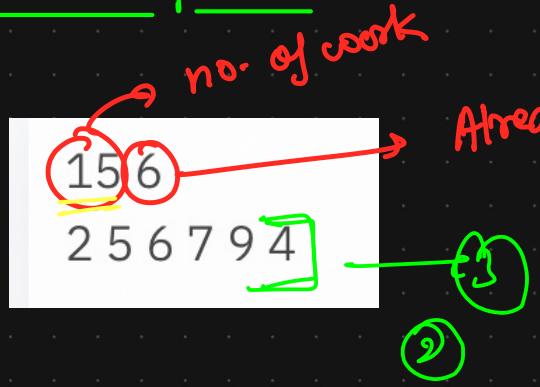
b ⟶ count 2 − count 1

⟶ separator.

We have to manage freq in value of hashmap.

Initial push [0 # 0 ⟶ 1]

Task Completion :→



① HashSet → already solved works

A B A B A B A B A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

A → 1, 8, 11, 13, 15] → Print

B → 3, 10, 12, 14] → True

② Make a loop from 1 to n. ← no. of task

If already solved
skip
↳ Alternate assign.