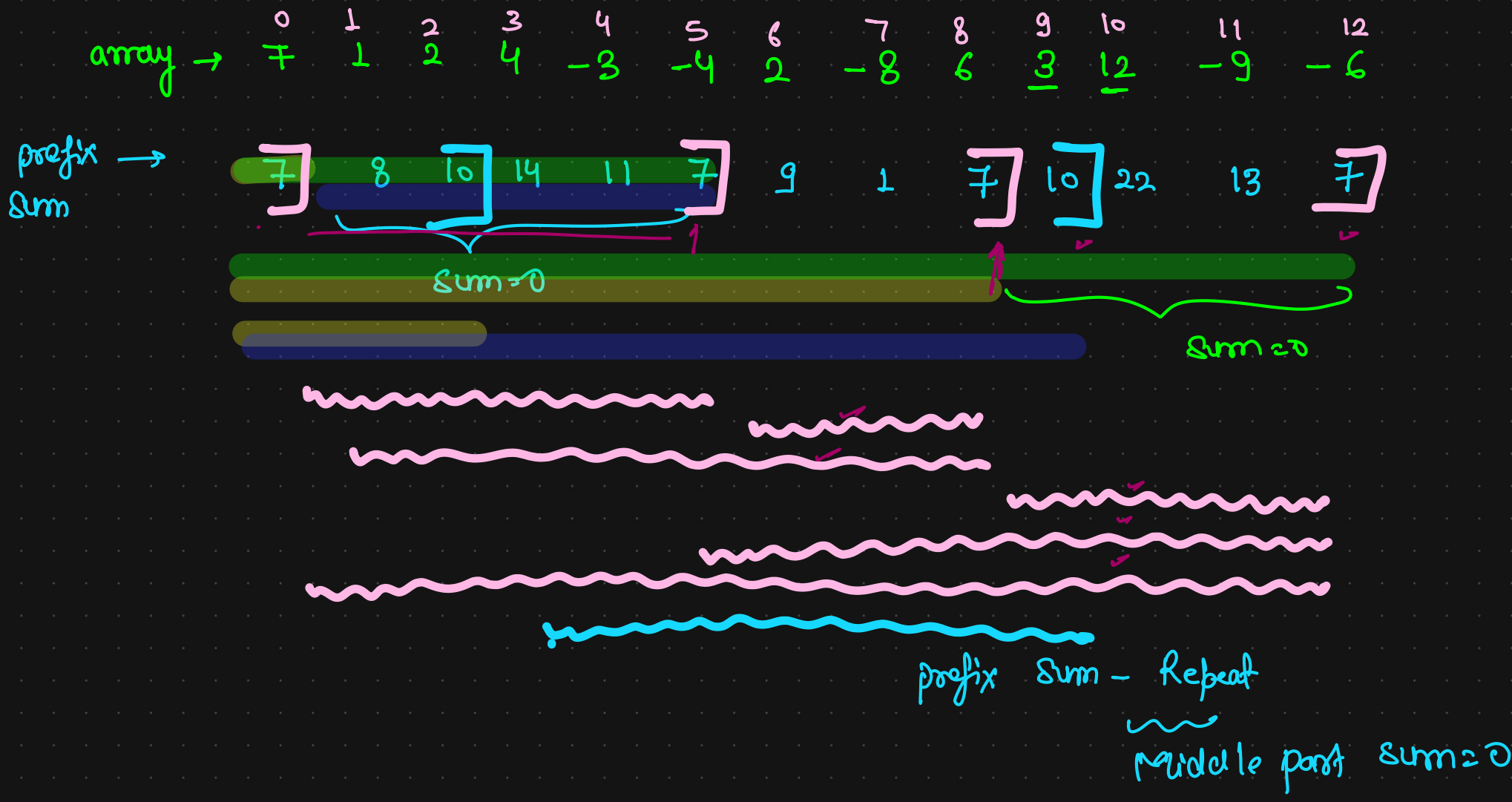
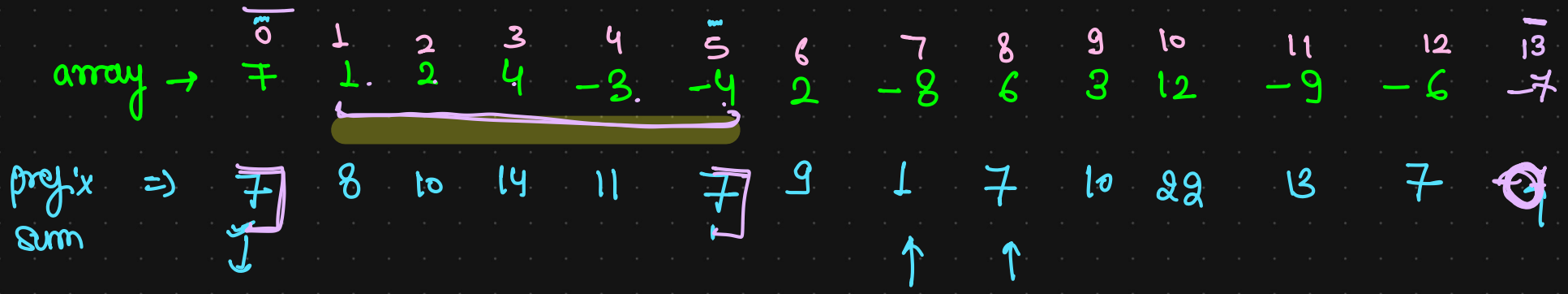
30th December 2021

- ~~L~~argest Subarray With Zero Sum → Easy tag-
- ~~C~~ount Of All Subarrays With Zero Sum → think about it
- Longest Subarray With Equal Number Of Zeroes And Ones
- Count Of Subarrays With Equal Number Of Zeroes And Ones
- Maximum Size Subarray Sum Equals K
- Count Of Subarrays Having Sum Equals To K

Largest Subarray with 0 Sum :-





Hashmap → prefix sum (key) vs Index of first occurrence of prefix sum.

→ Subarray start from 0 to i

→ Intentionally 0 → -1

7 → 0

8 → 1

10 → 2

14 → 3

11 → 4

9 → 6

1 → 7

22 → 10

13 → 11

length = 0 2 8 12 (max length of subarray having sum is 0)

= ↳ 0 sum subarray

```
if( prefix sum doesn't exist){
    make change in H.map.
} else {
⇒ length = max(length, i - map.get(psum))
    //
}
```

0	1	2	3	4	5
1	3	-2	-3	4	6

length = 3

psum =

1	4	2	0	4	10

0 → -1
1 → 0
4 → 1
2 → 2
0 → 3
10 → 5

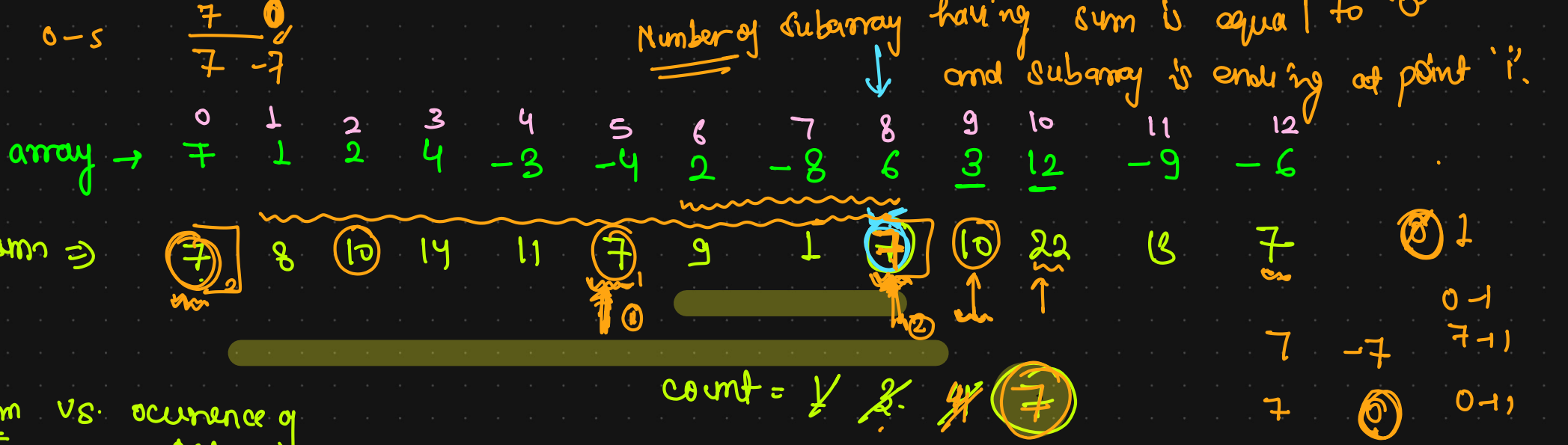
If prefix sum is '0', consider subarray from 0 index to 'i' index.

3 - (-1) = 4 length.

0	1	2	3	4	5	6	7	8	9	10	11	12
7	1	2	4	-3	-4	2	-8	6	3	12	-9	-6

Count of all Subarray with zero sum:

Number of subarray having sum is equal to '0' and subarray is ending at point 'i'.



psum vs. occurence of psfx.

*0 → 1
7 → 1 & 3
8 → 1
10 → 2
14 → 1
11 → 1
9 → 1
1 → 1
22 → 1
13 → 1

1 2 4 -3 -4

2 -8 6

1 2 4 -3 -4 2 -8 6

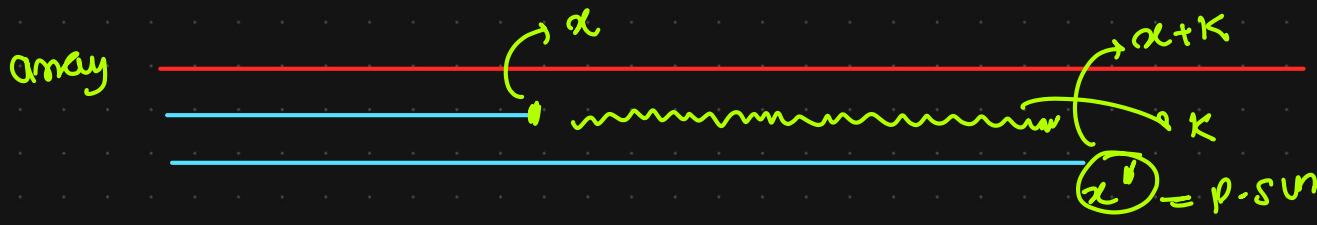
4 -3 -4 2 -8 6 3

8 12 -9 -6

2 -8 6 3 12 -9 -6

1 2 4 -3 -4 2 -8 6 3

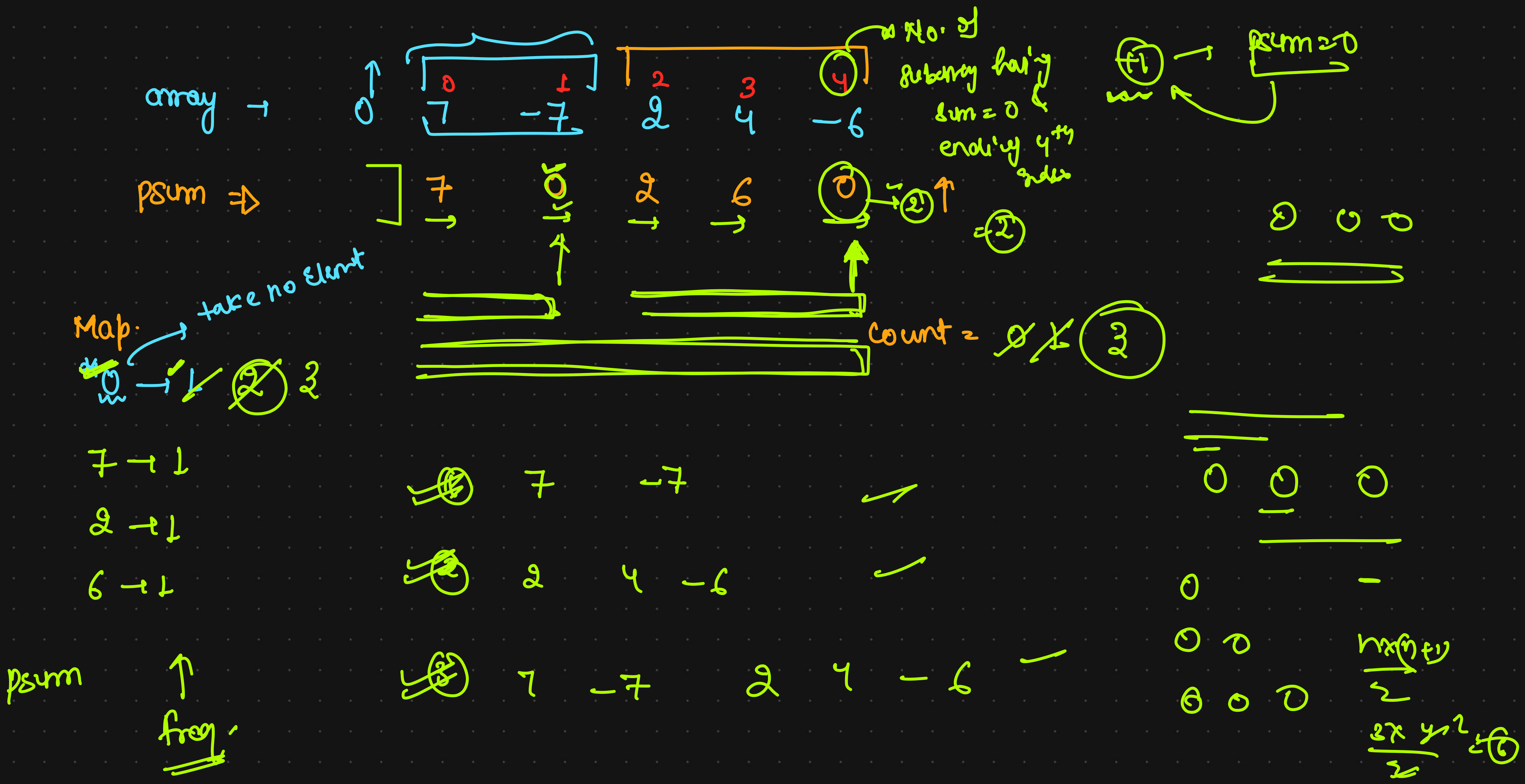
12 -9 -6



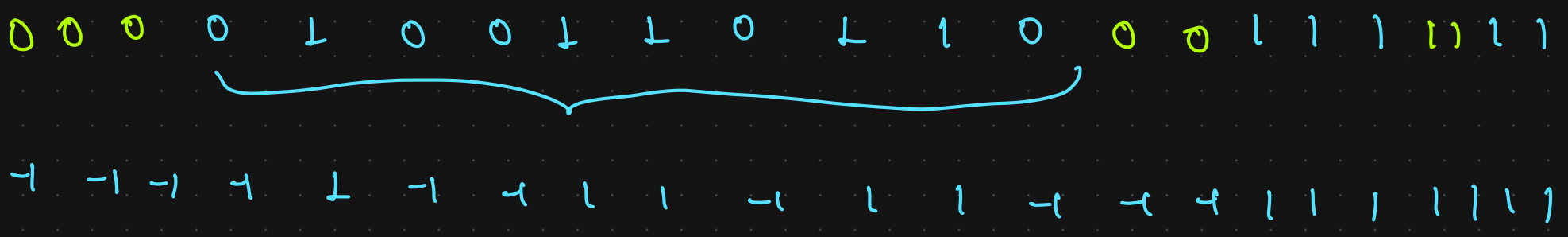
what prefix sum we search in H.Map

x'-k

k=0


array → 0 | 7 -7 2 4 -6
0 1 2 3 4
psum ⇒ 7 0 2 6 0
No. of subarray having sum = 0 & ending 4th index
psum = 0
take no element
Map
0 → 1 2 3
7 → 1
2 → 1
6 → 1
count = 0 1 3
0 7 -7
0 2 4 -6
0 7 -7 2 4 -6
psum
freq
n×(n+1)/2
3×4/2 = 6

Longest Subarray with Equal Number of 0s and 1s :→



Approach →

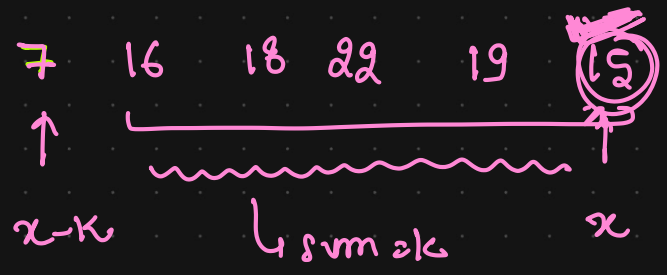
① change 0 → −1

② Solve for subarray having sum = 0

Count Subarray with equal no. of 0s and 1s.

Maximum Size Subarray Sum equal to k: → k = 8

	0	1	2	3	4	5	6	7	8	9	10	11	12
array →	7	9	2	4	-3	-4	6	-8	10	3	12	-1	-6
psum:	7	16	18	22	19	15		15					



length.

15 - 1 = 14

7 → 0

16 → 1

18 → 2

22 → 3

19 → 4

15 →

k = 0

length comparison →

put →

length comparison. →

if psum-k exist in hashmap

if(psum not exist in map) {

map.put(psum, i);

}