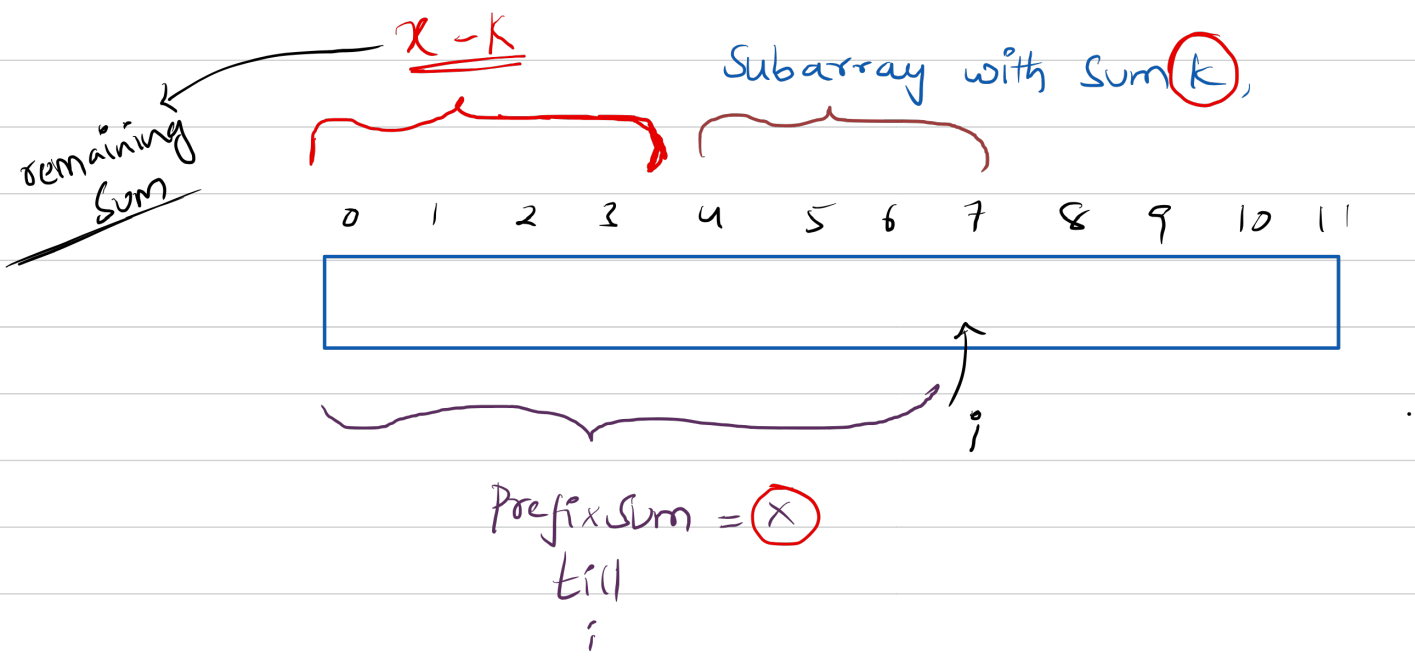
Subarray Sum equals K

Let us assume,



for a Subarray ending at index i with the prefix Sum x, if we remove the part with the prefix sum x-k, we will be left with the part whose sum is equals to k.

* There may exist multiple subarrays with the prefix sum (x-k). So, the number of subarrays with sum k that we can generate from the entire subarray ending at index i,

is exactly equal to the number of subarrays with the prefix sum $(X-K)$, that we can remove,

So, we keep the occurances of the prefix sum of the subarrays using HashMap

Steps

① initialise map ⟶ {0-1}

↳ Because,

Example:-

[1, 1, 1], K=2

The possible subarray can be [1, 1, 1] [1, 1, 1]

If you observe first subarray, the prefix Sum = 0, so by chance if any subarray that results (sum == k) starts with begining index, we need to increment the count

So ⟹ we initialize map as {0, 1}

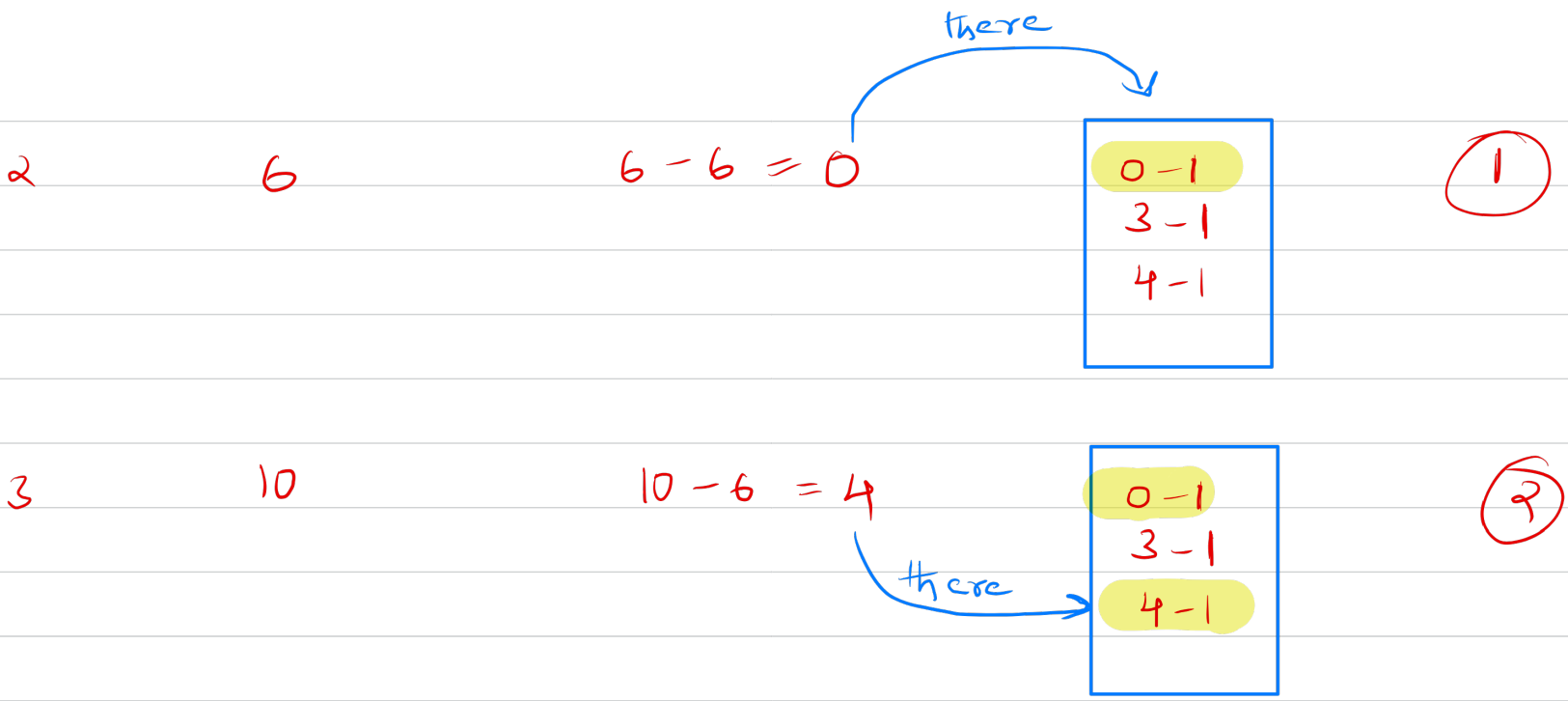
* Run a loop,

↳ On Each index

1. add current element → to prefixSum
2. Calculate (x-k)Sum
3. add occurance of (x-k) to our answer
4. Store current prefixSum in map by increasing its occurance by 1

Example: arr = {3, 1, 2, 4}, k=6

i	x	x-k	map	count
0	3	3-6 = -3 (not there)	0-1, 3-1	0
1	4	4-6 = -2 (not there)	0-1, 3-1, 4-1	0




```
map.put(0, 1);

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

{
    prefixSum += arr[i]

    int remainingSum = prefixSum - k;

    cnt += map.getOrDefault(remainingSum, 0);

    map.put(prefixSum, map.getOrDefault(prefixSum, 0) + 1);
}
```

Add the occurance of prefixSum (X - K) if available