**Find largest/smallest subarray with sum k in Given Array**

Step 1 : Calc the sum upto current index .

Step 2 : if the current sum is X and there exist an subarray with sum " K " then there also exist an sub array of X - K.

Step 3 : So by this reverse eng , we can find al the sub array of sum k .

Step 4 : When there is an sub array found in the middle then calc length by i - existing sum index.

Step 5 : If the current sum not exist in the sub array then put the sum along with the index .

Step 6 : else skip because we need the longest suppose there already exist the same sum before if we re updating it now then the sub array length will be decreased.

Map<Integer , Integer> map = new HashMap<>();

int n = nums.length;

int sum = 0;

int max = 0;

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

sum += nums[i];

if(sum == k) max = Math.max(max , i + 1);

if(map.containsKey(sum - k)){

max = Math.max(max , i - map.get(sum - k));

}

if(!map.containsKey(sum)) map.put(sum , i);

}

return max;