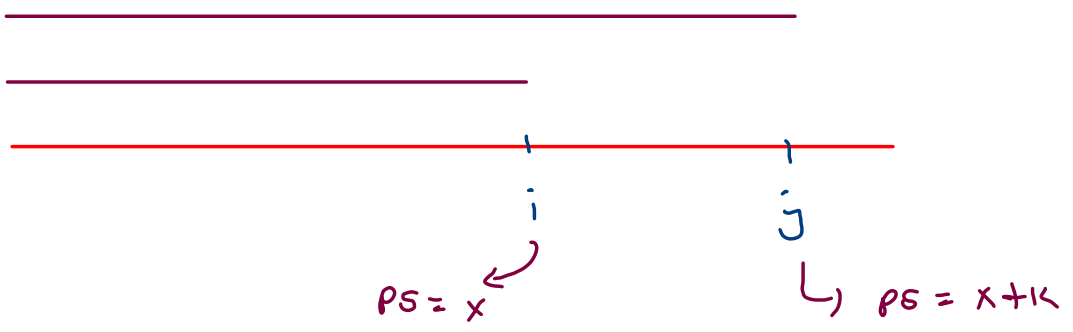
Maximum Size Subarray Sum Equals K

k = 15

arr: 10, 5, 2, 7, 1, 9



conclude, i+1 to j sum = k

arr: 10 5 2 7 1 9 -9 K = 15

0 1 2 3 4 5 6

ps 10 15 17 24 25 34 25

0 -> -1 24 -> 3

10 -> 0 25 -> 4

15 -> 1 34 -> 5

17 -> 2

6

~~4~~

olen = ~~0~~ ~~2~~

map -> ps vs j

```
6
342718
9
```

3 4 2 7 1 8

0 1 2 3 4 5

k = 9

ps 3 7 9 16 17 25

map ps vs ji

0 → -1
3 → 0
7 → 1
9 → 2
16 → 3
17 → 4
25 → 5

olen = ~~0~~ 3

560. Subarray Sum Equals K

Medium 12127 378 Add to List Share

Given an array of integers `nums` and an integer `k`, return *the total number of subarrays whose sum equals to* `k`.

```
for(int i=0; i < nums.length;i++) {
    ps += nums[i];

    if(map.containsKey(ps-k) == true) {
        oans += map.get(ps-k);
    }

    int nf = map.getOrDefault(ps,0) + 1;
    map.put(ps,nf);
}
```

map (ps vs freq)

k = 9

0 → 1
3 → 1
7 → 2
9 → 2
16 → 2

13 → 2

	3	4	2	7	-3	-6	2	4	3
	0	1	2	3	4	5	6	7	8
ps	3	7	9	16	13	7	9	13	16

oans = 1 + 1 + 1 + 2

0 to 2
2 to 3
0 to 6
6 to 8
2 to 8

```
arr[] = {10, 2, -2, -20, 10},
k = -10
```

10₀ 2₁ -2₂ -20₃ 10₄ k = -10

PS 10 12 10 -10 0

map 0 → 2 -10 → 1

10 → 2

12 → 1

ans = 1 + 2

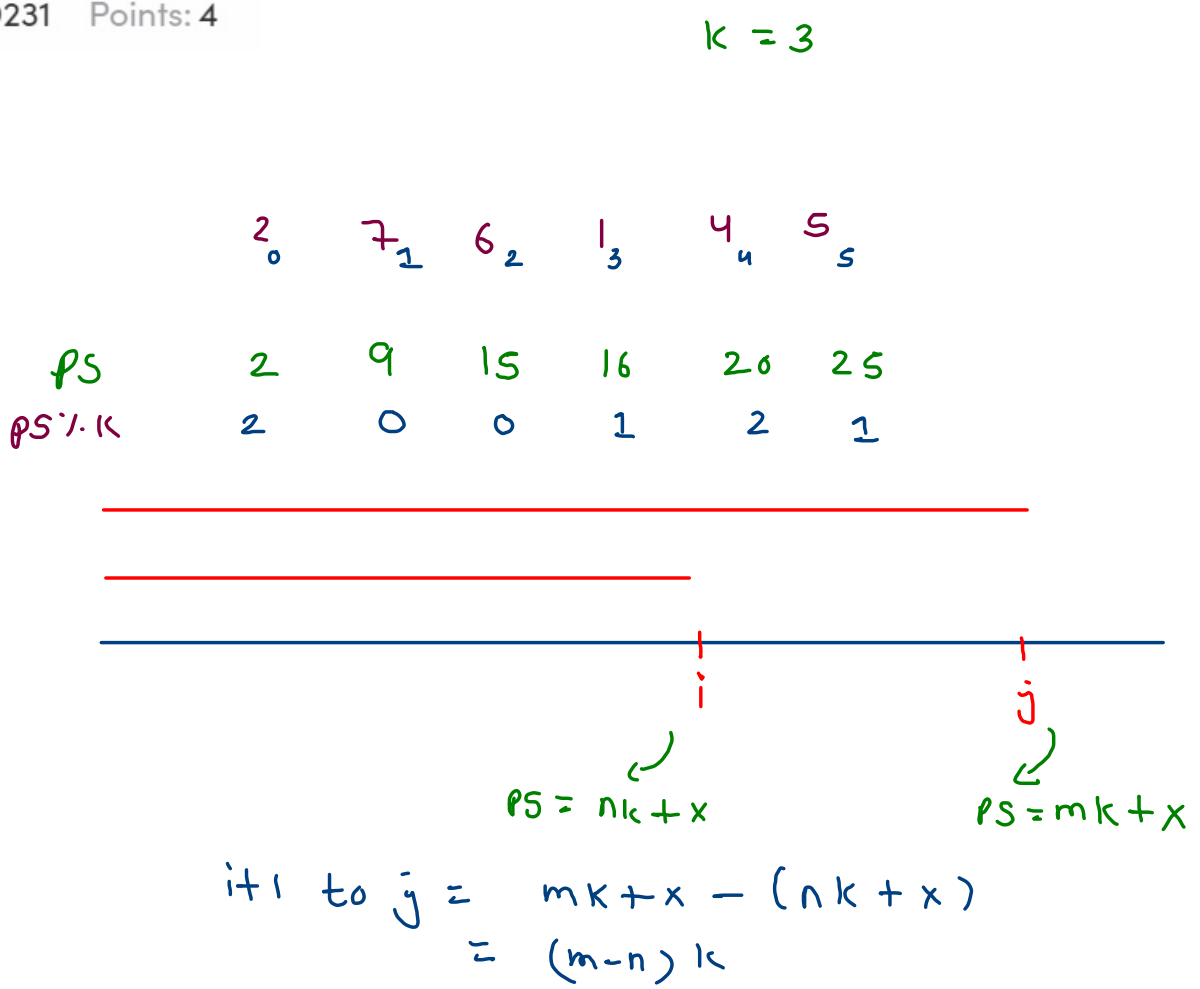
Longest subarray with sum divisible by K

Medium Accuracy: 43.42% Submissions: 9231 Points: 4

Input:
A[] = {2, 7, 6, 1, 4, 5}
K = 3
Output: 4
Explanation: The subarray is {7, 6, 1, 4} with sum 18, which is divisible by 3.

Example 2:

Input:
A[] = {-2, 2, -5, 12, -11, -1, 7}
K = 3
Output: 5
Explanation:
The subarray is {2,-5,12,-11,-1} with sum -3, which is divisible by 3.



Input:
A[] = {-2, 2, -5, 12, -11, -1, 7}
K = 3

K = 3

```
for(int i=0; i < arr.length;i++) {
    ps += arr[i];
    int rem = ps % k;

    if(rem < 0) {
        rem += k;
    }

    if(map.containsKey(rem) == true) {
        int len = i - map.get(rem);
        olen = Math.max(len,olen);
    }
    else {
        map.put(rem,i);
    }
}
```

	-2_0	2_1	-5_2	12_3	-11_4	-1_5	7_6
ps	-2	0	-5	7	-4	-5	2
(ps%k) rem	1	0	1	1	2	1	2

map
rem vs i

0 → (-1)
1 → 0
2 → 4

olen = ~~2~~ ~~3~~ 5

Longest Subarray With Equal Number Of 0s 1s And 2s

0 0 1 2 2 0 1 0 2 1 0 1

i

0's = x

1's = y

2's = z

j

0's = a = x+k — (1)

1's = b = y+k — (2)

2's = c = z+k — (3)

there are equal no. of 0, 1, 2 in i+1 to j

$c_0 - c_1$ @ $c_1 - c_2$

(1) - (2)

(2) - (3)

$a - b = x - y$ — (i)

$b - c = y - z$ — (ii)

```java
for(int i=0; i < arr.length;i++) {
    if(arr[i] == 0) {
        c0++;
    }
    else if(arr[i] == 1) {
        c1++;
    }
    else {
        c2++;
    }

    String state = (c0-c1) + "@" + (c1-c2);

    if(map.containsKey(state) == true) {
        int len = i - map.get(state);
        olen = Math.max(olen,len);
    }
    else {
        map.put(state,i);
    }
}
```

map

0 @ 0 → -1

1 @ 0 → 0

2 @ 0 → 1

1 @ 1 → 2

1 @ -1 → 4

2 @ -1 → 5

0 @ 1 → 7

	0_0	0_1	1_2	2_3	2_4	0_5	1_6	1_7
c0	1	2	2	2	2	3	3	3
c_1	0	0	1	1	1	1	2	3
c_2	0	0	0	1	2	2	2	2
state	1@0	2@0	1@1	1@0	1@-1	2@-1	1@0	0@1

olen = ~~0~~ ~~2~~ 6