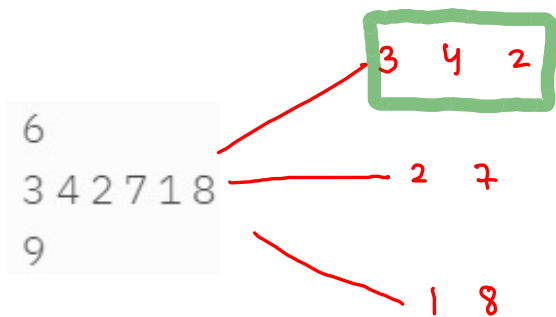
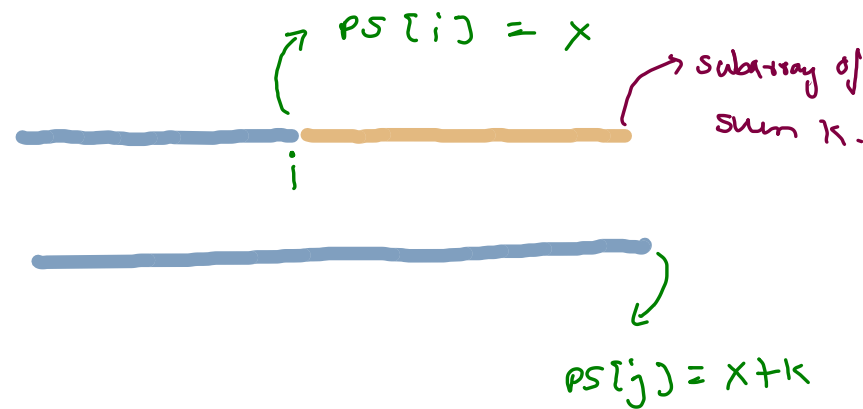


Maximum Size Subarray Sum Equals K



ans \rightarrow 3

$T: O(n)$



3_0 4_1 2_2 7_3 1_4 8_5 2_6 -9_7 5_8 6_9

ps:

3 7 9 16 17 25 27 18 23 29 $0 \rightarrow (-1)$

$3 \rightarrow (0)$

$7 \rightarrow 1$

$9 \rightarrow 2$

$16 \rightarrow 3$

$17 \rightarrow 4$

$25 \rightarrow 5$

$27 \rightarrow 6$

$18 \rightarrow 7$

$23 \rightarrow 8$

$29 \rightarrow 9$

hashmap

ps vs first index

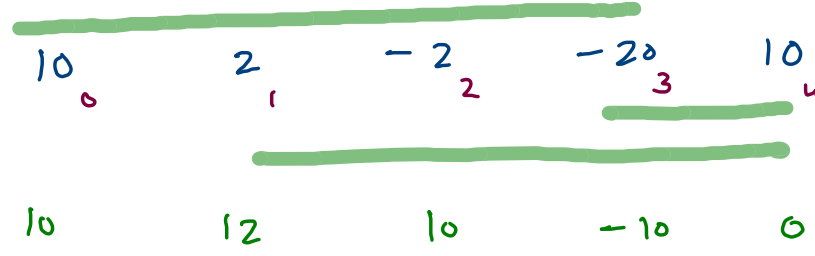
find : $ps - k$

$k = 9$

ans = ~~3~~ 5

```
= { 10, 2, -2, -20, 10 };
-10;
```

$$k = -10$$



ps

10

12

10

-10

0

0 \rightarrow 1

10 \rightarrow 2

12 \rightarrow 1

-10 \rightarrow 2

$$0 - (-10) = 10$$

$$\text{count} = 1 + 2$$

hash map

ps vs freq

```
HashMap<Integer,Integer>map = new HashMap<>(); //ps vs freq
int count = 0;
int ps = 0;

map.put(0,1);

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

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

    if(map.containsKey(ps) == false) {
        map.put(ps,1);
    }
    else {
        int nf = map.get(ps) + 1;
        map.put(ps,nf);
    }
}
```

$$k = -10$$



ps

10

12

10

-10

0

ps vs first

0 → (-1)

10 → 0

12 → 1

-10 → 3

ans = 4

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

    if(map.containsKey(ps-k) == true) {
        int len = i - map.get(ps-k);

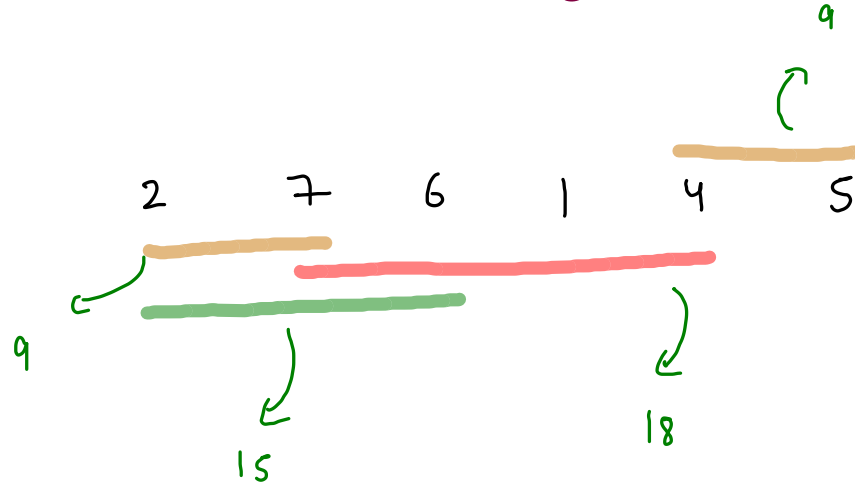
        if(len > ans) {
            ans = len;
        }
    }

    if(map.containsKey(ps) == false) {
        map.put(ps,i);
    }
}
```

Longest Subarray With Sum Divisible By K

$K = 3$

6
2 7 6 1 4 5
3



hashmap : sum vs first idx

\hookrightarrow

\hookrightarrow

$PS \div K$

$$PS[i] = K * n + x$$

$$PS[j] - PS[i] = Km + x - (Kn + x) = K(m - n)$$

$$PS[j] = K * m + x$$

$$k = 3$$

1₀

2₁

7₂

-4₃

5₄

-10₅

6₆

1₇

-4₈

-5₉

PS

1

3

10

6

11

1

7

8

4

-1

rem

1

0

1

0

2

1

1

2

1

2

0 \rightarrow (-1)

1 \rightarrow 0

2 \rightarrow 4

ans = ~~4~~ ~~5~~ ~~6~~ 8

hashmap

rem vs first idx

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

1 0 0 2 1 0 0 1 2 1 0

c_0 0 1 2 2 2
 c_1 1 1 1 1 2
 c_2 0 0 0 1 1

$c_0 = x$
 $c_1 = y$
 $c_2 = z$
 prev

$k = 2$

$c_0 = x + k$
 $c_1 = y + k$
 $c_2 = z + k$
 cur

$(c_1 - c_0) @ (c_2 - c_1)$
 0 @ -1

2

1

3

$$(c_1 - c_0) @ (c_2 - c_1)$$

	1	0	0	2	1	0	0	1	2	1	0
c_0	0	1	2	2	2	3	4	4	4	4	5
c_1	1	1	1	1	2	2	2	3	3	4	4
c_2	0	0	0	1	1	1	1	1	2	2	2
state	1@-1	0@-1	-1@-1	-1@0	0@-1	-1@-1	-2@-1	-1@-2	-1@-1	0@-2	-1@-2

$$\begin{matrix} x \\ y \\ z \end{matrix} \left. \begin{matrix} \\ \\ \end{matrix} \right\} \begin{matrix} y-x \\ z-y \end{matrix}$$

$$\begin{matrix} x+k \\ y+k \\ z+k \end{matrix} \left. \begin{matrix} \\ \\ \end{matrix} \right\} \begin{matrix} y-x \\ z-y \end{matrix}$$

ans $\rightarrow 3/6$

HashMap

$\langle \text{String}, \text{Integer} \rangle$
 $\downarrow \quad \quad \downarrow$
 state first occ
 $c_1 - c_0 @ c_2 - c_1$

Ans = ~~6~~

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

    if(map.containsKey(st) == true) {
        int len = i - map.get(st);

        if(len > ans) {
            ans = len;
        }
    }
    else {
        map.put(st,i);
    }
}
```

		1 ₀	0 ₁	0 ₂	2 ₃	1 ₄	0 ₅	0 ₆	1 ₇	2 ₈	1 ₉	0 ₁₀
-1												
C ₀ 0	0	1	2	2	2	3	4	4	4	4	5	
C ₁ 0	1	1	1	1	2	2	2	3	3	4	4	
C ₂ 0	0	0	0	1	1	1	1	1	2	2	2	
st 0@0	1@-1	0@-1	-1@-1	-1@0	0@-1	-1@-1	-2@-1	-1@-2	-1@-1	0@-2	-1@-2	

$$\begin{array}{lcl}
 C_0 \rightarrow x & & \\
 C_1 \rightarrow y & \left. \begin{array}{l} \\ \\ \end{array} \right\} & \begin{array}{l} y-x \\ z-y \end{array} \\
 C_2 \rightarrow z & &
 \end{array}$$

$$\begin{array}{lcl}
 x + k_0 & \left. \begin{array}{l} \\ \\ \end{array} \right\} & \begin{array}{l} y-x + (k_1 - k_0) \\ z-y + (k_2 - k_1) \end{array} \\
 y + k_1 & & \\
 z + k_2 & &
 \end{array}$$

$k_1 = k_0$

$k_2 = k_1$

$k_0 = k_1 = k_2$