

Q → Find the length of longest subarray with  $\text{sum} = 0$ .

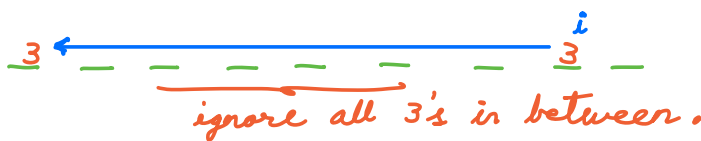
No subarray → Ans = -1

$A = [3 \ 3 \ 4 \ -5 \ -2 \ 2 \ 1 \ -3 \ 3]$   
 $P = [3 \ 6 \ 10 \ 5 \ 3 \ 5 \ 6 \ 3 \ 6]$

→  $(P[i] == P[j]) \Rightarrow \text{subarray } (i+1) \text{ to } j \text{ has sum} = 0$

$[L \ R] \rightarrow R - L + 1$

$A = [3 \ 3 \ 4 \ -5 \ -2 \ 2 \ 1 \ -3 \ 3]$   
 $P = [3 \ 6 \ 10 \ 5 \ 3 \ 5 \ 6 \ 3 \ 6]$



$A = [3 \ 3 \ 4 \ -5 \ -2 \ 2 \ 1 \ -3 \ 3]$   
 $P = [3 \ 6 \ 10 \ 5 \ 3 \ 5 \ 6 \ 3 \ 6]$

$V[P[i]] \rightarrow \text{store first index where } P[i] \text{ is present.}$

HashMap

<  $P[i]$ , first index >

ans = -1

for  $i \rightarrow 0 \text{ to } (N-1)$

if ( $\text{hm.containsKey}(P[i])$ )

$l = \text{hm.get}(P[i]) + 1$

$r = i$

$\text{ans} = \max(\text{ans}, r - l + 1)$

else

$A = [3 \ 3 \ 4 \ -5 \ -2 \ 2 \ 1 \ -3 \ 3]$   
 $P = [3 \ 6 \ 10 \ 5 \ 3 \ 5 \ 6 \ 3 \ 6]$

$i$	Map	length
0	3 → 0	
1	6 → 1 ✓	
2	10 → 2	
3	5 → 3	

subarrays

$[1 \ 4] \rightarrow 4$   
 $[1 \ 7] \rightarrow 7$   
 $[5 \ 7] \rightarrow 3$   
 $[2 \ 6] \rightarrow 5$   
 $[2 \ 8] \rightarrow 7$   
 $[7 \ 8] \rightarrow 2$   
 $[4 \ 5] \rightarrow 2$   
 Ans = 7

$P[i] == P[j]$

$i$	$j$
0	4
3	5
1	6
0	7
1	8

```

[ ] hm.put(P[i], i)
0 1 2 3 4
A = [4 -3 -1 2 -2]
P = [4 1 0 2 0]
    ↑
    Ans = 5 ✓

```

```

4 [1 4] → 4
5 [4 5] → 2
6 [2 6] → 5
7 [1 7] → 7
8 [2 8] → 7

```

```

ans = -1
P[0] = A[0]
if (P[0] == 0) ans = 1
for i → 1 to (N-1)
    P[i] = P[i-1] + A[i]
    if (P[i] == 0)
        ans = i+1
for i → 0 to (N-1)
    if (hm.containsKey(P[i])) ← ✓
        l = hm.get(P[i]) + 1 = 1+1 = 2
        r = i = 2
        ans = max(ans, r - l + 1)
    else
        hm.put(P[i], i)
return ans

```

```

0 1 2 3
A = [3 8 0 7]
P = [3 11 11 18]
    ↑
A = [0 3 8]
P = [0 3 11]
map
3 → 0
11 → 1
18 → 3

```

TC =  $O(N)$

SC =  $O(N)$

2 - 2 + 1 = 1 ✓

Q → Given an array containing only 0's & 1's.  
Find max length of a subarray which contains equal no. of 0's & 1's.

```

0 1 2 3 4 5 6 7
A = [0 1 0 0 1 1 0 1] Ans = 8

```

```

0 1 2 3 4 5 6 7 8
A = [0 1 0 0 0 1 1 0 1] Ans = 8

```

max length subarray with sum = 0

```

A = [0 1 0 0 0 1 0 0 1 1 0 0]
    [-1 1 -1 -1 -1 1 -1 -1 1 1 -1 -1]

```

replace 0 with -1.

equal 0's & 1's  $\Rightarrow$  equal -1's & 1's  $\Rightarrow$  sum = 0 ✓

H.W  $\rightarrow$  Count the # subarrays with sum = 0 for given integer A[].

10:25 PM

Q  $\rightarrow$  Given an integer array A & an integer K,  
Hyland check if there exist a pair (i, j) s.t

$$i \neq j \text{ \& \& } A[i] + A[j] = K$$

A = [ 8 9 1 -2 4 5 11 ]

K = 20      A[1] + A[6] = 9 + 11 = 20      Ans = true

K = 9      A[0] + A[2] = 8 + 1 = 9  
            A[4] + A[5] = 4 + 5 = 9  
            A[3] + A[6] = -2 + 11 = 9      }      Ans = true

K = 0  $\rightarrow$  Ans = false

A = [ 8 -6 0 1 0 3 ]

$\rightarrow$  K = 16    i = 0, j = 0    X

K = 0      A[2] + A[4] = 0 + 0 = 0      Ans = true      A[0] + A[2] = 16 ✓

Bruteforce  $\rightarrow$   $\forall i, j$  check if  $A[i] + A[j] = K$ .

```
for i  $\rightarrow$  1 to (N-1)
  for j  $\rightarrow$  0 to (i-1)
    if (A[i] + A[j] == K)
      return true
return false
```

$i \neq j$      $\because A[i] + A[j] = A[j] + A[i]$   
 $\therefore$  only consider  $i < j$  or  $i > j$

TC =  $O(N^2)$     SC =  $O(1)$

$\forall i$  find A[j] s.t  $A[i] + A[j] = K \Rightarrow$   $A[j] = K - A[i]$

$\forall i$ , check if  $K - A[i]$  is present in input.

Hashset

Solution Steps → 1) for  $i \rightarrow 0$  to  $(N-1)$  } Insert all elements  
 $hs.add(A[i])$  } in hashset.

2) for  $i \rightarrow 0$  to  $(N-1)$  }  
 if (  $hs.contains(K - A[i])$  ) } check if  $K - A[i]$  is  
 return true } present in input  $\forall i$ .  
 return false

$A = [ \overset{0}{8} \overset{1}{-6} \overset{2}{0} \overset{3}{1} \overset{4}{0} \overset{5}{3} ]$

$K = 16$  (16-8) present in input → Hashset ✓ ( $i=j$ )  
 Ans = true ✗

$\forall i$ , check if  $K - A[i]$  is present in input from index 0 to  $(i-1)$ .

for  $i \rightarrow 1$  to  $(N-1)$

for  $j \rightarrow 0$  to  $(i-1)$   
 if  $(A[i] + A[j]) == K$   
 return true

return false

$\forall$  index  $i \rightarrow$  only elements from  
 index 0 to  $i-1$  should  
 be part of Hashset.

$A = [ \overset{0}{9} \overset{1}{7} \overset{2}{420} \overset{3}{-2} \overset{4}{0} \overset{5}{9} ]$

$K = 7$

i       $K - A[i]$       Hashset

0       $7 - 9 = -2$       { }

1       $7 - 7 = 0$       { 9 }

2       $7 - 420 = -413$       { 9, 7 }

3       $7 - (-2) = 9$       { 9, 7, 420 } ✓ return true

for  $i \rightarrow 0$  to  $(N-1)$

if (  $hs.contains(K - A[i])$  )  
 return true

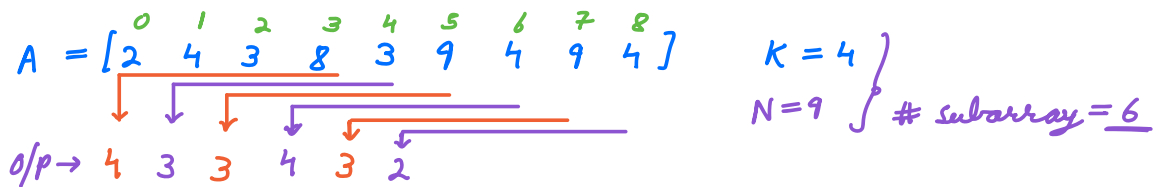
$hs.add(A[i])$

return false

$TC = O(N)$

$SC = O(N)$

Q → Given an integer array  $A$  & an integer  $K$ .  
 Calculate the number of distinct elements in  
 every subarray of size  $K$ .



Bruteforce → ∀ subarray of size K, count # distinct elements.

# subarrays of size K =  $\underline{N - K + 1}$        $O(K)$  { Add all elements in Hashset, Ans = Hashset size.

$$TC = O((N - K + 1) * K)$$

$$= \underline{O(N * K)}$$

$$K = \frac{N}{2}$$

$$O((N - K + 1) * K) = \underline{O(N^2)}$$

$$(N - \frac{N}{2} + 1) * \frac{N}{2} \approx \frac{N}{2} * \frac{N}{2} = \underline{\frac{N^2}{4}}$$

$$SC = \underline{O(K)}$$

Fixed size subarray → sliding window

$A = [2, 4, 3, 8, 3, 9, 4, 9, 4]$        $K = 4$       + Hashset

Remove → i      Add → i-K

i	Add	Remove	Hashset	Size
0-3			{2, 4, 3, 8}	4
4	A[4] = 3	A[0]	{4, 3, 8}	3
5	A[5] = 9	A[1]	{3, 8, 9}	3
6	A[6] = 4	A[2] = 3	{8, 9, 4}	3

o/p → 4 3 3 4 3 2

HashMap

$A[i] \rightarrow$  freq of  $A[i]$

{ Hashset → Remove an element completely, irrespective of how many times it was added. ✓

$A = [2, 4, 3, 8, 3, 9, 4, 9, 4]$        $K = 4$

i	Add	Remove	HashMap → delete	Size
0-3			{2:1, 4:1, 3:1, 8:1}	4
4	A[4]	A[0]	{4:2, 3:1, 8:1}	3
5	A[5]	A[1]	{3:2, 8:1, 9:1}	3

6	A[6]	A[2]	{3:1, 8:1, 9:1, 4:1}	4
7	A[7]	A[3]	{3:1, 9:2, 4:1}	3
8	A[8]	A[4]	{9:2, 4:2}	<u>2</u>

Sliding Window + HashMap

TC = O(N)   SC = O(K)

H.W. → Pseudocode

---