

## Agenda

- 1. First Non Repeating Element
- 2. Pair sum
- 3. Count of Pair Sum
- 4. Check if there exists a subarray with sum = 0
- 5. Check if there exists a subarray with sum = K



**< Question > :** Given N elements. Find the first non-repeating element.

$$N = 6$$

[ 1 2 3 1 2 5 ]

ans : 3

$$N = 8$$

[ 4 3 3 2 5 6 4 5 ]

$$N = 7$$

[ 2 6 8 4 7 2 9 ]



Iterate over the array & for every element we can check if it is repeating or not.

T.C :  $O(N^2)$

S.C :  $O(1)$



Idea-2 Using Hashmap

{ 1 2 3 1 2 5 }  
↑  
ans : 3

ele	Freq
1	2
2	2
3	1
5	1

- Create hashmap < ele, freq >
- Iterate over the array again & check for 1st ele having freq = 1.



&lt;/&gt; Code

```
int FirstNonRepeating( int[ ] A) {
```

```
    Hashmap< Int, Int > hm;
```

```
    for( i=0 ; i< A.length ; i++ ) {  
        if( hm. search( A[i] ) == true ) {  
            int f = hm. get( A[i] );  
            int nf = f++;  
            hm. insert( A[i], nf );  
        }  
        else {  
            hm. insert( A[i], 1 );  
        }  
    }
```

```
    for( i=0 ; i< n ; i++ ) {  
        if( hm. get( A[i] ) == 1 )  
            return A[i];  
    }  
    return -1;
```

T.C : O(N)  
S.C : O(N)

{ 3, 3, -3, -3, 1, 2 }

3	3
1	
2	

2
1
3



< Question > : Given an arr[ N ] and K.

Check if there exists a pair ( i , j ) such that,  $\text{arr}[i] + \text{arr}[j] = K$  &&  $i \neq j$

arr →

8	9	1	-2	4	5	11	-6	4
0	1	2	3	4	5	6	7	8

$K = 6 \quad \text{arr}[2] + \text{arr}[5]$  True

$K = 22$  False

$K = 8 \quad \text{arr}[4] + \text{arr}[8]$  True

Quiz :

0	1	2	3	4	5
3	5	1	2	1	2

$K = 7 \quad \text{arr}[1] + \text{arr}[3]$  True

$K = 10$  False



BF Idea Generate all pairs & check if any pair has sum = k.

```
for(i=0; i<n; i++) {  
    for(j=i+1; j<n; j++) {  
        if (arr[i]+arr[j] == K)  
            return true;  
    }  
}  
return false;
```

T.C :  $O(N^2)$

S.C :  $O(1)$



OR

$$A[i] + A[j] = k$$

$$A[j] = k - A[i]$$

$$x = A[i]$$

$$y = A[j]$$

```

for(i=0 ; i<n ; i++) {
    x = A[i];
    for(j=i+1 ; j<n ; j++) {
        if (arr[j] == k - x) {
            return true;
        }
    }
}
return False;

```



## Idea - 2 Using hashset.

$$A = \{8, 9, 2, -2, 4, 5, 11\} \quad K = 18$$

↑

$x$	8	9				
$y$	10	9				
$(K-x)$						

Not Correct

True.

8	-2	11
9	4	
2	5	

$$A = \{8, 9, 2, -2, 4, 10, 11\} \quad K = 18$$

$x$	8	9	2	-2	4	10			
$y$	10	9	16	20	14	8			
$(K-x)$									

8	-2	
9	4	
2		

Return True.

HS: {0 to i-1}



Idea: Using Hashmap.

- Check if  $y$  is present in hm.
- If ( $x == y$ )  
check for freq of  $y$ .  
IF  $> 1 \rightarrow$  Return True.

HashMap<ele, freq>

ele freq

8	-1
9	-1
2	-1
-2	-1
4	-1
5	-1
11	-1

</> Code HashSet approach

```
boolean PairSum(int []A, int K) {
```

```
    HashSet<Int> hs ;
```

```
    for (i=0 ; i < A.length ; i++) {
```

```
        x = A[i] ;
```

```
        y = K - x ;
```

```
        if (hs.search(y) == true)
```

```
            return true ;
```

```
        else
```

```
            hs.add(x) ;
```

```
y
```

```
return false ;
```

T.C : O(N)

S.C : O(N)



## QUESTION Count the number of pairs with sum = K

Given an arr[n], count number of pairs such that

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

Note that (i, j) and (j, i) considered as same.

arr[ ] → [ 2 5 2 5 8 5 2 8 ]      K = 10

$$\{0, 4\}$$

$$\{2, 4\}$$

$$\{6, 4\}$$

$$\{0, 7\}$$

$$\{2, 7\}$$

$$\{6, 7\}$$

$$\{1, 3\}$$

$$\{1, 5\}$$

$$\{3, 5\}$$

ans : 9

## QUIZ

{ 3 5 1 2 1 2 }      K = 3

$$\{2, 3\}$$

$$\{2, 5\}$$

$$\{3, 4\}$$

$$\{4, 5\}$$

ans : 4



## Idea -1      Use hashmap

$\text{arr[ ]} \rightarrow [ 2 \ 5 \ 2 \ 5 \ 8 \ 5 \ 2 \ 8 ]$        $K = 10$

ele	freq
2	$x \ 2 \ 3$
5	$x \ 2 \ 3$
8	$x \ 2$

$\text{count} = \emptyset ;$   
 $1 + 2 + 2 + 1 + 3 = 9$

```
int CountTargetSum(int [] arr, int K){
```

```
    count = 0 ;
```

```
    Hashmap< Int , Int > hm ;
```

```
    for ( i=0 ; i< A.length ; i++ ) {
```

```
        x = A[i] ;
```

```
        y = K - x ;
```

T.C : O(N)

S.C : O(N)

```
        if ( hm.search(y) == true ) {
```

```
            count += hm.get(y) ;
```

```
            if ( hm.search(x) == true )
```

```
                hm.insert( x , hm.get(x)+1 ) ;
```

```
            else
```

```
                hm.insert( x , 1 ) ;
```

y

return count ;



</> Code      HashSet Approach — Willn't work.

```
int CountTargetSum(int [] arr, int K) {
```

```
    count = 0;
```

```
    HashSet<Int> hs;
```

```
    for (i=0 ; i < A.length ; i++) {
```

```
        x = A[i];
```

```
        y = K - x;
```

```
        if (hs.search(y) == true)
```

```
            count++;
```

```
        else
```

```
            hs.add(x);
```

```
y
```

```
return count;
```

```
y
```

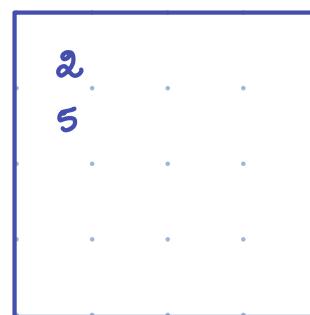
arr[ ] → [ 2 5 2 5 8 5 2 8 ]

K = 10



x	2	5	2	5	8
y	8	5	8	5	2

count = 1 + 1





## QUESTION

# Subarray with Sum 0

**< Question > :** Given an array of N elements. Check if there exists a subarray with sum equal to 0.

( $1 \leq N \leq 10^5$ )

$N = 11$       arr  $\rightarrow$  [ 0 1 2 3 4 5 6 7 8 9 ]      2 2 1 -3 4 3 1 -2 -3 2      True.



### BF Idea

Generate all the subarrays & check if any subarray has  
sum = 0.

3 nested loops

T.C:  $O(N^3)$

S.C:  $O(1)$

Prefix Sum

T.C:  $O(N^2)$

S.C:  $O(N)$

Carry forward

T.C:  $O(N^2)$

S.C:  $O(1)$



## Idea-2 Using psum[] &amp; hashset.

N = 10

0	1	2	3	4	5	6	7	8	9
2	2	1	-3	4	3	1	-2	-3	2

psum[]: [ 2 4 5 2 6 9 10 8 5 7 ]

$$\begin{aligned} \text{Sum}[i:j] &\Rightarrow \text{psum}[j] - \text{psum}[i-1] = 0 \\ &\Rightarrow \text{psum}[j] = \text{psum}[i-1] \end{aligned}$$

Observation 1: Repeating values in psum[] denotes subarray sum = 0.

0	1	2	3
-2	-1	3	5

psum[] → [ -2 -3 0 5 ]

Observation 2: If 0 is present in psum[], we have subarray sum = 0.

2 ways to handle this:

- If  $\text{psum}[i] = 0$  Return True
- Explicitly add 0 in hashset.

Dry Run :

N = 10

0	1	2	3	4	5	6	7	8	9
2	2	1	-3	4	3	1	-2	-3	2

psum[]: [ 2 4 5 2 6 9 10 8 5 7 ]

2  
4  
5



&lt;/&gt; Code

```
boolean subarrayzero( int[ ] arr ) {
```

```
    // Create psum[ ]
```

```
    HashSet< Integer > hs ;
```

T.C : O(N)

S.C : O(N)

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

```
        if ( . hs. search ( psum [ i ] ) == true || psum [ i ] == 0 )
```

```
            return true ;
```

```
        else
```

```
            hs. add ( psum [ i ] ) ;
```

```
}
```

```
    return false ;
```

```
}
```

Use carry forward to save prefix sum array space.



## QUESTION

# Subarray with Sum $\rightarrow K$

< Question > : Given an array[ N ]. Check if there is a subarray with sum = k

( $1 \leq N \leq 10^6$ )

arr[ ]  $\rightarrow$  [ 0 1 2 3 4 5 6 7 8 ]  
[ 2 3 9 -4 1 5 6 2 5 ]

K = 11      True      (5,6)      (2,5)  
K = 10      True      (0,4)

## QUIZ

arr[ ]  $\rightarrow$  [ 0 1 2 3 4 ]  
[ 5 10 20 100 105 ]      K = 110  
False



### BF.Idea

Generate all the subarrays & check if any Subarray has sum = K.

3 nested loops

T.C:  $O(N^3)$

S.C:  $O(1)$

Prefix Sum

T.C:  $O(N^2)$

S.C:  $O(N)$

Carry forward

T.C:  $O(N^2)$

S.C:  $O(1)$



## Idea-2 Using psum[] & hashset.

$$\text{Sum}[i:j] \Rightarrow \underbrace{\text{psum}[j]}_x - \underbrace{\text{psum}[i-1]}_y = k$$
$$\Rightarrow x - y = k$$
$$\Rightarrow y = x - k$$

arr[ ] → [ 2 3 9 -4 1 5 6 2 5 ]       $k = 11$

psum[ ] → [ 2 5 14 10 11 16 22 24 29 ]



$x$  . . 2 . 5 . 14 . 10 . 11  
 $y$  . . -9 . 6 . 3 . -1 . 0  
 $(x-k)$

↓  
Return True.

0
2
5
14
10



&lt;/&gt; Code

```
int SubarraySumK( int[ ] arr, int K) {
```

```
    HashSet< Int > hs;
```

```
    int sum = 0;
```

```
    hs.add(0);
```

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

```
        sum += A[i];
```

```
        x = sum;
```

```
        y = x - K;
```

```
        if (hs.search(y) == true)
```

```
            return true;
```

```
        else
```

```
            hs.add(x);
```

```
    }
```

T.C : O(N)

S.C : O(N)

```
}
```























