

## Today's Quote:



## Today's content

- pair sum = K
- pair difference = K
- Distinct elements in every window of len = K.

Q: Given N array elements, 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, 7, 8, 5, 2 }

$K = 11$  : YES, (4, 8)

$K = 6$  : YES (0, 3), (2, 5)

$K = 22$  : NO

ideas → Consider all the pairs & check if their sum = K.

pseudo-code :

```
for( i=0 ; i < N ; i++ ) {
    a = arr[i] , b = K-a
    for( j=i+1 ; j < N ; j++ ) {
        if( arr[j] == b ) {
            return true
        }
    }
}
```

T.C →  $O(N^2)$   
S.C →  $O(1)$

$$\begin{aligned} \underbrace{\text{arr}[i]}_a + \underbrace{\text{arr}[j]}_b &= K \\ a + b &= K \\ \{ b = K-a \} \end{aligned}$$

$\text{arr}[1] = \{ 8, 9, 1, -2, 4, 5, 11, -6, 7, 8, 5, 9 \}$

- idea: → optimisation using HashSet.  
 → insert all the elements in HashSet.

$hs =$	8	9	1	-2	4
	5	11	-6	7	

<u><math>K=11</math></u>	$a$	$b = (K-a)$	<u><math>b</math> is present ?</u>
	8	3	No
	9	2	No
	1	10	No
	-2	13	No
	4	7	YES {return true}

<u><math>K=5</math></u>	$a$	$b = (K-a)$	<u><math>b</math> is present ?</u>
	8	-3	No
	9	-4	No
	1	4	YES {return true}

<u><math>K=-4</math></u>	$a$	$b = (K-a)$	<u><math>b</math> is present ?</u>
	8	-12	No
	9	-13	No
	1	-5	No
	-2	-2	YES {return true}

Wrong ans # No of -2's present in the array = 1.  
 {ans should be false}.

Idea 3. optimisation using Hashmap-

→ iterate on all elements & store elements in Hashmap.

arr[T] : {  
8 9 4 5 11 -6 7 5 -2 4  
0 1 2 3 4 5 6 7 8 9 }

hm →	8 → 1	-6 → 1
	9 → 1	7 → 1
	4 → 2	-2 → 1
	5 → 2	
	11 → 1	

Key → element  
value → frequency.

K=10

<u>a</u>	<u>b = (K-a)</u>
8	2
9	1
4	6
5	5

is b present in hm?

No

No

No

If ( $a == b$  && freq(a) > 1)  
return True

K=22.

<u>a</u>	<u>b = (K-a)</u>
8	1
9	13
4	18
5	17
11	11

is b present in hm?

No

No

No

No

If ( $a == b$  && freq > 1)

-b	-
7	-
5	-
-2	-
4	-

## pseudo code

```
boolean pairSum ( arr, N, K) {  
    Hashmap < integer, integer > hm;  
    // insert all the elements in hm { Todo }  
  
    for( i=0 ; i < N ; i++ ) {  
        a = arr[i] , b = K-a  
        if ( a != b && hm.search(b) == true ) { return true }  
        if ( a == b && hm[a] > 1 ) { return true }  
    }  
    return false;  
}
```

T.C  $\rightarrow O(N)$   
S.C  $\rightarrow O(N)$

### idea-4 Optimisation using HashSet

→ At  $i^{th}$  index, HashSet will contain Unique elements upto  $(i-1)^{th}$  index.

arr[T] : { $\underset{0}{8}, \underset{1}{9}, \underset{2}{4}, \underset{3}{5}, \underset{4}{11}, \underset{5}{-6}, \underset{6}{7}, \underset{7}{5}, \underset{8}{-2}, \underset{9}{4}\}$

<u>K=22</u>	<u>a</u>	<u>b = (K-a)</u>	<u>HashSet</u>	<u>is b present in hs?</u>
	8	14	{ - }	No
	9	13	{ 8 }	No
	4	18	{ 8, 9 }	No
	5	17	{ 8, 9, 4 }	No
	11	11	{ 8, 9, 4, 5 }	No

Code is now working  
for this edge case.

<u>K=10</u>	<u>a</u>	<u>b = (K-a)</u>	<u>HashSet</u>	<u>is b present in hs?</u>
	8	2	{ - }	No
	9	1	{ 8 }	No
	4	6	{ 8, 9 }	No
	5	5	{ 8, 9, 4 }	No
	11	-1	{ 8, 9, 4, 5 }	No
	-6	16	{ 8, 9, 4, 5, 11 }	No
	7	3	{ 8, 9, 4, 5, 11, -6 }	No
	5	5	{ 8, 9, 4, 5, 11, -6, 7 }	YES, return true.

## Final pseudo code :

```
boolean pairSum( arr, N, K) {  
    HashSet<integer> hs;  
    for( i= 0 ; i < N ; i++) {  
        a = arr[i] , b = K-a  
        if( hs.search( b) == true) {  
            return true  
        }  
        hs.insert( arr[i])  
    }  
    return false ;
```

T.C  $\rightarrow O(N)$   
S.C  $\rightarrow O(N)$

Q) Given N array elements, check if there exists a pair  $(i, j)$  such that their difference  $= K$  &  $i \neq j$

$$\text{arr} = \{ 5, 10, 3, 2, 50, 80 \}, K = 78.$$

<u><math>K=78</math></u>	<u><math>a</math></u>	<u><math>b = (a-K)</math></u>	<u>Hashset</u>	<u>is b present in hs?</u>
	5	-73	{ - }	No
	10	-68	{ 5 }	No
	3	-75	{ 5, 10 }	No
	2	-76	{ 5, 10, 3 }	No
	50	-28	{ 5, 10, 3, 2 }	No
	80	2	{ 5, 10, 3, 2, 50 }	YES. { return true }

$$\text{arr} = \{ 5, 10, 3, 80, 50, 2 \}, K = 78.$$

<u><math>a</math></u>	<u><math>b = (a-K) \text{    } (a+K)</math></u>	<u>Hashset</u>	<u>is b present in hs?</u>
5	-73, 83	{ - }	No
10	-68, 88	{ 5 }	No
3	-75, 81	{ 5, 10 }	No
80	2, 158	{ 5, 10, 3 }	No
50	-28, 128	{ 5, 10, 3, 50 }	No
2	-76, 80	{ 5, 10, 3, 80, 50 }	{ Yes } return true.

Difference of two no's = K.

$$\begin{aligned} a - b &= K \\ \Rightarrow b &= a - K \\ b - a &= K \\ \Rightarrow b &= a + K \end{aligned}$$

Pseudo-code:

```
boolean pairDifference ( arr, N, K ) {  
    HashSet<integer> hs;  
  
    for( i= 0 ; i < N ; i++ ) {  
        a = arr[i] , b1 = a-K , b2 = a+K  
        if ( hs.search(b1) || hs.search(b2) )  
            return true  
        hs.insert( arr[i] )  
    }  
    return false;  
}
```

T.C  $\rightarrow O(N)$   
S.C  $\rightarrow O(1)$

Q1 Given N elements. Calculate no. of distinct elements in every subarray of size K.

Eg: arr[10] : { 2, 4, 3, 8, 3, 9, 4, 9, 4, 10, 9 }

K=4

[0-3] : 4

[1-4] : 3

[2-5] : 3

[3-6] : 4

[4-7] : 3

[5-8] : 2

[6-9] : 3

idea: for every subarray of size = K,  
insert all the elements in hashset & get no.  
of distinct elements.

T.C -  $(N-K+1) \cdot K = O(N^2)$ , S.C  $\rightarrow O(N)$

$$\hookrightarrow K=1, (N-K+1) \cdot 1 = N$$

$$K=N, (N-N+1) \cdot N = N$$

$$K=\frac{N}{2}, \left(N-\frac{N}{2}+1\right) \cdot \left(\frac{N}{2}\right) = \frac{N}{2} \cdot \frac{N}{2} = \frac{N^2}{4} = O(N^2)$$

optimisation  $\rightarrow$  Sliding window + hashset.

arr[10] : { 2, 4, 3, 8, 3, 9, 4, 9, 4, 10, 9 }, k=4.

s. s.

Hashset

# no. of distinct ele =  
hash.size()

0 3

remove

add

{ 2, 4, 3, 8 }

4

1 4

arr[0]

arr[4]

{ 4, 3, 8 }

3

2 5

arr[1]

arr[5]

{ 3, 8, 9 }

3

3 6

arr[2]

arr[6]

{ 8, 9, 4 }

3 ✗ {wrong}

Note  $\rightarrow$  frequency of element is also important. { Use hashmap }

### Idea-2. Using Hashmap

$\text{arr} \rightarrow \{ 2, 4, 3, 8, 3, 3, 9, 4, 9, 4, 10 \}$ ,  $K=4$

<u><math>\frac{S}{E}</math></u>	<u><math>\frac{C}{E}</math></u>	<u>Hashmap</u>	<u>Size of hm.</u>
0 3	<u>remove</u>	<u>add</u>	4
1 4	$\text{arr}[0]$	$\text{arr}[4]$	3
2 5	$\text{arr}[1]$	$\text{arr}[5]$	3
3 6	$\text{arr}[2]$	$\text{arr}[6]$	4
4 7	$\text{arr}[3]$	$\text{arr}[7]$	3
5 8	$\text{arr}[4]$	$\text{arr}[8]$	2
6 9	$\text{arr}[5]$	$\text{arr}[9]$	3

## final pseudo code-

```
void distinctElements ( arr, N, K ) {  
    Hashmap < integer, integer > hm;  
    //insert first K-elements in hm.  
    for ( i = 0 ; i < K ; i++ ) {  
        if ( hm.search ( arr[i] ) == true ) {  
            hm[ arr[i] ] += 1  
        } else {  
            hm.insert ( arr[i], 1 );  
        }  
    }  
    print( hm.size() ) // no. of distinct elements in  
                      // first subarray of size = K
```

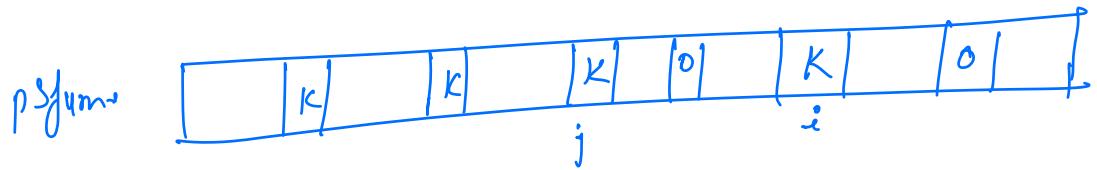
s=1, e=K

```
while ( e < N ) {  
    // get the subarray from s to e  
    remove arr[s-1] add arr[e]  
    hm[ arr[s-1] ] -= 1  
    if ( hm[ arr[s-1] ] == 0 ) { hm.delete ( arr[s-1] ) }  
    if ( hm.search( arr[e] ) == true ) { hm[ arr[e] ] += 1 }  
    else { hm.insert ( arr[e], 1 ) }  
    print ( hm.size() );
```

s++, e++

T.C  $\rightarrow O(N)$ , S.C  $\rightarrow O(K)$

Hint. Count of subarrays with sum = 0.

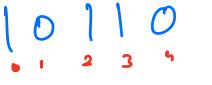


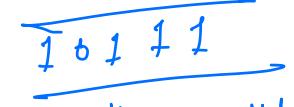
observation → We have 3 subarrays with sum = 0, ending at index  $i$

Colorful No.

$$\underline{3} \underline{2} \underline{4} \underline{5} \Rightarrow \{3, 2, 4, 5, 32, 24, 45, 324, 245, 3245\}$$

3	,	40
2	,	120
4		
5		
6		
8		
20		
24		

$s1 \rightarrow$   (22) reverse  
 $s2 \rightarrow$   (7)

$s1 \rightarrow$    
 $s2 \rightarrow$    
ans:   
↓ reverse this to get  
your final ans

ans:   $\Rightarrow$  (29)

str "hello - world ---"

rev  $\Rightarrow$  "dlrow -- olleh"