

## Hashing 2

Hello Everyone



### Today's Agenda :

- 1) Pair sum K
- 2) Count no. of pairs with sum == k
- 3) Subarray with sum == k
- 4) Distinct elements in every window of size k.

Q1) Given an array A of size N and K.  
Check if there exists a pair  $(i, j)$  such that  
 $A[i] + A[j] = K$  and  $i \neq j$ .

Eg.

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

$K=6$

Output : True

Quiz : {3, 5, 1, 2, 1, 2}

$K=7$

Output : True

Quiz : {3, 5, 1, 2, 1, 2}

$K=10$

Output : False

Brute Force : Run two loops.

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

TC:  $O(n^2)$   
SC:  $O(1)$

$$a+b = K$$

$$b = K-a$$

a  
↓

K=7

{ 3, 5, 1, 2, 1, 2 }

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

$$a = A[i], b = K - a$$

```
for(j=i+1; j<n; j++)  
{
```

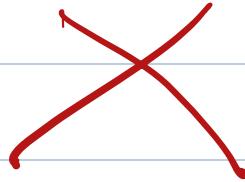
if ( $A[j] == b$ )

{  
 return true;  
}

}  
return false;

Idea : 2

Using HashSet



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

▷ Insert all the element inside HashSet.

for (i=0; i < n; i++)  
d

a = A[i]

b = K - a;

if (hs.search(b) == true)

{  
    return true;  
}

}  
return false;

i

{3, 5, 1, 2, 1, 2}     K=10     (i == j)

8	9	1	
-2	4	5	11
-6			

HS-

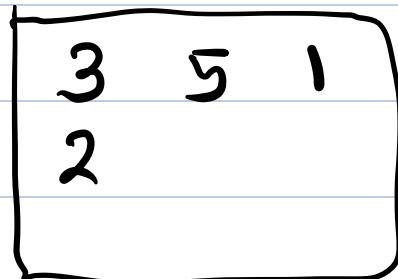
▷ Insert all the ele in set.

3	5
1	2

# Create HashSet on the go | While iterfy.

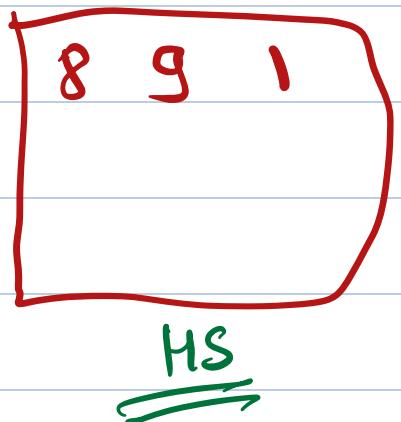
i  
 $\{3, 5, 1, 2, 1, 2\}$        $K=10$

False



i  
0 1 2 3 5 6 7 8 9  
 $\boxed{8} \quad 9 \quad 1 \quad \boxed{-2} \quad 4 \quad 5 \quad 11 \quad -6 \quad 4$        $K=6$

True



Code:

HashSet < Integer > hs;

for (i=0; i<n; i++)  
    {

        a = A[i];

        b = K - a;

        if (hs.search(b) == true)

            {

                return true;

        hs.insert(a);

        return false;

TC: O(n)

SC: O(n)

Qviz : Count pairs  $\text{arr}[i] + \text{arr}[j] == K$ ,  $i != j$   
 $A = [3, 5, 1, 2, 1, 2]$   $K=3$

Output :

Index 1	Index 2
$2 \rightarrow 1$	$3 \rightarrow 2$
$2 \rightarrow 1$	$5 \rightarrow 2$
$3 \rightarrow 2$	$4 \rightarrow 1$
$4 \rightarrow 1$	$5 \rightarrow 2$

Q.2) Count the no. of pairs with sum =  $K$ ,  
note :  $i != j$

$A \rightarrow$ 

2	5	1	2	5	8	5	2	8
---	---	---	---	---	---	---	---	---

 $K=10$

1, 3

3, 5

Ans = 9

0, 4

0, 7

2, 4

2, 7

1, 5

6, 7

4, 6

# Optimisation

i

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

K=10

2 → X X 3

5 → X 2 3

8 → 1

$$\text{Ans} = 0 + 1 + 2 + 2 \\ + 1 + 3$$

$$= 9$$

Code:

HashMap<Integer, Integer> hm;

ans = 0

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

a = A[i];      b = K - a;

if (hm.search(b) == true)

{

    ans = ans + hm.get(b);

    // insert curr ele inside map.

    if (hm.search(a) == true)

{

        hm.insert(a, hm.get(a) + 1);

    }

    else

{

        hm.insert(a, 1);

}

return ans;

TC: O(n)

SC: O(n)

## \* SubArray with sum K

{5, 10, 20, 100, 105}

Q.3) Given an array and K. Check if there exists a pair  $(i, j)$  such that  $A[i] - A[j] = K$  and  $i > j$ .

	0	1	2	3	4	5	6	7	8	$i$
	2	5	14	10	11	16	22	24	29	

$K = 9$

$$i = 2 (14) \quad j = 1 (5)$$

$$14 - 5 = 9$$

should be on right side of b-

$$a - b = K$$

HashSet<Int32> hs;

for each a, b should be

$$= a - K$$

for ( $i = 0$ ;  $i < N$ ;  $i++$ ) {  
 4, 5, 14, 10, 11, 16, 22, 24, 25}

$b = A[i] - K;$

    if ( $hs.\text{search}(b) == \text{true}$ )

        return true;

    hs.insert( $A[i]$ );

return false;

$i$

$K = 9$

4 5

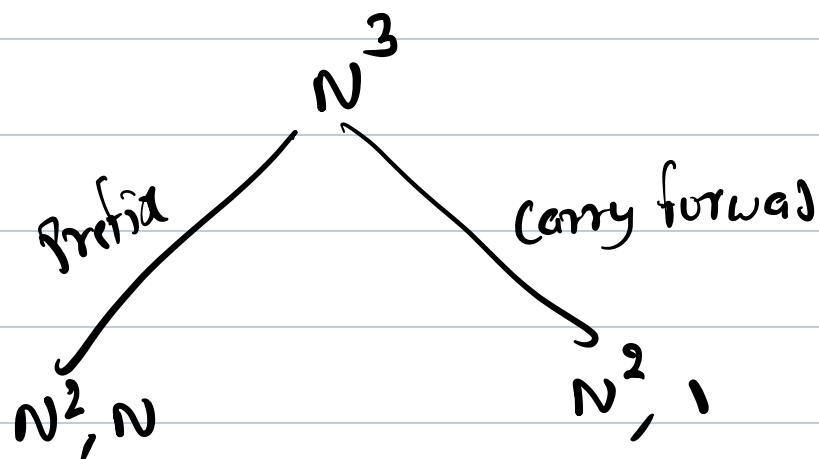
True

Actual question : Given an array. Check if there exists subarray with sum = k

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

K = 11

Bruteforce : Consider all the subarrays and for every subarray iterate and find the sum.



## Idea 2:

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

0	1	2	3	5 <sup>T</sup>	6 <sup>T</sup>	7 <sup>T</sup>	8	9
---	---	---	---	----------------	----------------	----------------	---	---

PF =

2	5	14	10	11	16	22	24	29
---	---	----	----	----	----	----	----	----

K=11

i

I am looking for pair whose indices diff == K  
 $i > j$

2	5	14	D
10	11	16	
22	24	29	

1) Create a PF sum array.

2) HashSet<integers> hs

```

hs.set(0);
for(i=0; i<n; i++)
{
    b = PF[i] - K
    if(hs.search(b) == true)
        return true;
    hs.insert(PF[i]);
}

```

TC: O(n)  
SC: O(n)

return False,

## Scenerio

**Scaler Academy** wants to make learning better by looking at the different number of topics students work on within a duration. They plan to boost engagement if the number of topics covered within a certain period falls below expectations.

## Problem

Given an array **A** of **N** integers, denoting the sequence of topics a learner interacted with for a module, and a number **B** denoting the window size. Your mission is to determine the count of different topics the learner engaged within each window of size **B**.

Eg.  $A \rightarrow \{1, 2, 1, 3, 5, 2, 1\}$   $B = 3$

Output :  $\{2, 3, 3, 3, 3\}$

$A \rightarrow \{1, 2, 4, 3, 1, 2, 1\}$   $B = 4$

Output :  $\{4, 4, 4, 3\}$

Q.4) find the count of distinct elements in every window of size K.

$A \rightarrow \{1, 2, 1, 3, 4, 2, 3\}$   $K = 4$

Output:  $\{3, 4, 4, 3\}$

**Brute force:**  $\Rightarrow$  Iterate on every window of size K and insert every element inside

Hashset.

$\Rightarrow$  The size of hashset is going to determine the no. of unique element for that window.

$$(N - K + 1) * K$$

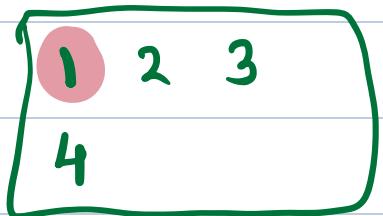
$$K = 1 \quad (N - 1 + 1) * 1 = N$$

$$K = N \quad (N - N + 1) * N = N$$

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

Idea 2: Sliding Window + Hashset

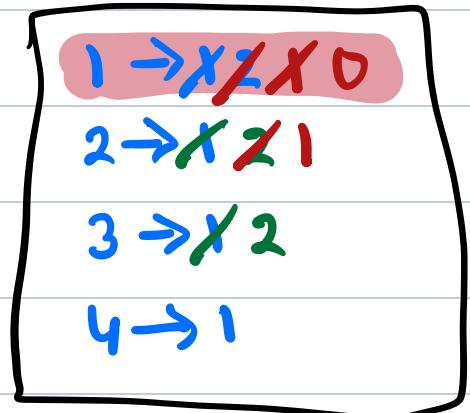
$A[] = \{1, 2, 1, 3, 4, 2, 3\}$   $K=4$



Idea 3: Hashmap + Sliding window

$A[] \rightarrow \{1, 2, 1, 3, 4, 2, 3\}$   $s$   $e$   $K=4$

ans:  $\{3, 4, 4, 3\}$



## Code:

```
HashMap<Integer, Integer> hm;
for(i=0; i<K; i++)
{
    if (hm.search(A[i]) == true)
    {
        hm.insert(A[i], hm.get(A[i]) + 1);
    }
    else
    {
        hm.insert(A[i], 1);
    }
}
```

int ans[N-K+1]  
ans[0] = hm.size()

$$s = 1 \quad e = K$$

while (e < N)

|| operate on new element of the window

```
if (hm.search(A[e]) == true)
{
    hm.insert(A[e], hm.get(A[e]) + 1);
}
```

hm.insert(A[e], hm.get(A[e]) + 1);

else  
{  
    hm.insert(A[i], i);  
}  
}

Operate on ele to be discarded in the new window ( $s-1$ )

val = hm.get(A[s-1]);  
hm.insert(A[s-1], val - 1);

If (val - 1 == 0)  
{  
    hm.remove(A[s-1]);  
}  
}

ans[s] = hm.size();

stt;  
ett;

}

$T_C \rightarrow O(N - k + k)$   
 $= O(n)$

$SC \rightarrow O(\kappa)$