

Hashing 2 : Problems

Question 1 :- Given arr[N] & K, check if there exists a pair (i, j), such that

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

Ex :- Index :- 0 1 2 3 4 5 6 7 8

array :- 8 9 1 -2 4 5 11 -6 4

if $K = 6 \rightarrow \text{arr}[2] + \text{arr}[5] : \text{True}$

$K = 22 \rightarrow \text{False}$

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

Quiz 1 :- check if there exists a pair (i, j) such that,
 $\text{arr}[i] + \text{arr}[j] == K \quad \& \quad i \neq j$ in the given array :

$A = [3, 5, 1, 2, 1, 2] \quad \& \quad K = 7.$

↳ True

Quiz 2 :- check if there exists a pair (i, j) such that,
 $\text{arr}[i] + \text{arr}[j] == K \quad \& \quad i \neq j$ in the given array :

$A = [3, 5, 1, 2, 1, 2] \quad \& \quad K = 10.$

↳ False

Brute force

→ check all possible pair

Ex:- array of length 4

	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

← All pairs

observation → ① Diagonal is invalid because of
 $i \neq j$

② you need to consider either
one of two ↗.

PSEUDO CODE

for ($i \rightarrow 0$ to $n-2$)

 for ($j \rightarrow i+1$ to $n-1$) {

 3

↳ OPTIMIZATION

IDEA1

Index :- 0 1 2 3 4 5 6 7 8

array :- 8 9 1 -2 4 5 11 -6 4

$$\begin{aligned} \text{arr}[i] &\leftarrow \text{arr}[j] \\ A + B &= K \\ \Rightarrow B &= K - A \end{aligned}$$

Let, $\underline{K = 8}$

- ① Insert all elements in Hashset
- ② Iterate 8 & check if complement exists

Day Run

<u>first (arr[i])</u>	<u>complement arr[j] = K - arr[i]</u>	<u>K = 8</u>
8	-1	
9	7	
1	10	
-2	4	
4	4	→ True
5		
11		
-6		
4		

Set

8
9 1 -2
4 5 11
-6 4

PSEUDO CODE

TODO → //Insert all the elements in set
 HashSet <Integer> set;

for (i → 0 to n-1) {

 other = K - arr[i];

 if (set. search (other)) {

 return true;

 return False;

Edge Case

Ex:- arr[] → 5, 7, 9, 2, 3

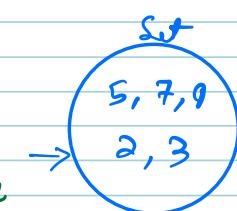
K=4

arr[i]

5
7
9
2

K - arr[i]

-1
-3
-5
2



Flow:- we are not storing freq..

IDEA-2: ↳ correction using HashSet

array :- 8 9 1 -2 4 5 11 -6 4
 ↑
 other
 $K - arr[i]$

Ex1 :- $K = 9$ first
 $arr[i]$)

8
9
1
:
:

1
0
8
:
:

→ True

Set

8, 9

Ex2 Ex :- $arr[] \rightarrow 5, 7, 9, 2, 3$

$K = 4$ $arr[i]$)

$K - arr[i]$

5
7
9
2
3

-1
-3
-5
2
1

5, 7
9, 2,
3

PSEUDO CODE

HashSet <Integer> set;

for (i → 0 to n-1) {

 other = $K - arr[i]$;

 if (set. search (other)) {

 return True;

 set. add (arr[i]);

 }

TC → O(N)
SC → O(N)

return False;

IDEAS

→ Using Hashmap

↳ TODO as H.W.

Ques 3 Count pair (i, j) such that,

$\text{arr}[i] + \text{arr}[j] = k$ & $i \neq j$ in the given array

$$A = [3, 5, 1, 2, 1, 2]$$

$$\& k = 3$$

$$\hookrightarrow \text{Ans} = 4$$

Question 2 :- Given an $\text{arr}[n]$, count no. of pairs such that

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

Ex :- $k = 10$

$$\text{Indexes} = 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$

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

Ans Pairs $\rightarrow (0, 4) (1, 3) (2, 4) (3, 5) (6, 2) (0, 7)$
 $(1, 5) (2, 7) (4, 6)$

$$\hookrightarrow [9]$$

Let , $k = 10$

$$\text{arr}[i]$$

$$2$$

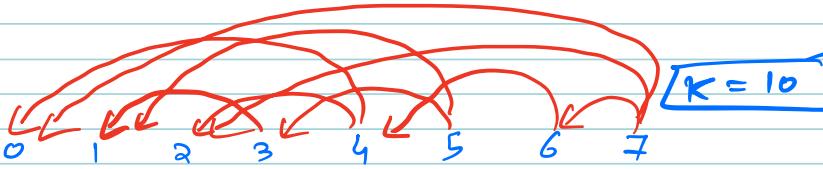
$$k - \text{arr}[i]$$

$$8 \searrow 2$$

observation → whenever we are on $\text{arr}[i]$, we will try to check the count of complements.

Dry Run

Indexes =



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

$\text{arr}[i]$

2

5

2

5

8

5

2

8

$K - \text{arr}[i]$

8

5

8

5

2

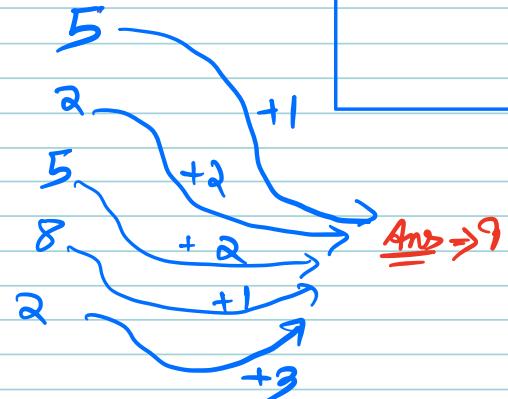
8

2

8

HashMap

$2 \rightarrow X^2 Z^3$
 $5 \rightarrow X^2 Z^3$
 $8 \rightarrow 1$



PSEUDO CODE

```
ans = 0;  
HashMap<Int, Int> map;  
for ( i → 0 to n-1) {  
    other = k - arr[i];  
    if ( map. check (other) ) {  
        ans += map.get (other);  
    }  
    if ( map. check (arr[i]) ) {  
        map [arr[i]] += 1;  
    } else {  
        map [arr[i]] = 1;  
    }  
}  
return ans;
```

TC → O(N)

SC → O(N)

Question 3 :- Given an array $arr[n]$ check if there exists a subarray with sum = k

Ex

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

y , $k = 11 : [2 3 9 -4 1] [5 6]$

$k = 10 : [2 3 9 -4]$

$k = 15 : [-4 1 5 6 2 5]$

Ques 4 :- check if there exist a subarray with sum = 110 in the given array?

$$A = [5, 10, 20, 100, 5]$$

↳ No

Brute force

↳ check all possible subarray sum & if its k return True.

↳ OPTIMIZATION

IDEA :- [using Prefix Sum]

Indexes	0	1	2	3	4	5	6	7	8
arr[9]	2	3	9	-4	1	5	6	2	5
Prefix Sum	2	5	14	10	11	16	22	24	29

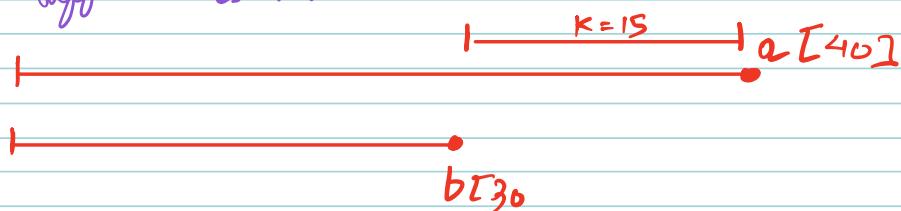
Real life Example

In Jan start

Account
5 lakh
↓
(3 lakh)
Save
8 lakh

In Feb start

So, we just need to find two points
whose diff is k .



IDEA :-

Diff in 2 Points = k

$$a - b = k$$

$$\Rightarrow b = a - k$$

Ex:- $K = 12$

Indexes	0	1	2	3	4	5	6	7	8
arr[9]	2	3	9	-4	1	5	6	2	5

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

2	$b = a - k$
2	-10
5	-7
14	2 → True.
10	

Hashset

2, 5

maths

$$\text{sum}(i-j) = Pf[r_j] - Pf[r_{i-1}]$$

$$K = Pf[r_j] - Pf[r_{i-1}]$$

$$Pf[r_{i-1}] = Pf[r_j] - K$$

↑ ↑
b a

PSEUDO CODE

$Pf = 0$
Hashset < int > set;

```

set.add(0);
for (i → 0 to n-1) {
    Pf = Pf + Ar[i];
    other = Pf - K;
    if (set.search(other)) {
        return True;
    }
    set.add(Pf);
}
return False;

```

Edge Case

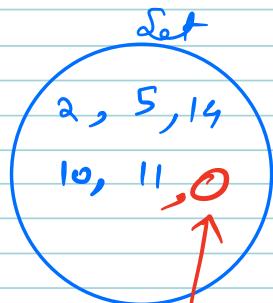
Ex:- arr[] = [2 3 9 -4 1]

PF = [2 5 14 10 11]

$$\sum K = 11$$

2	$b = a - k$
2	-9
5	-6
14	3
10	-1
11	0

→ return False



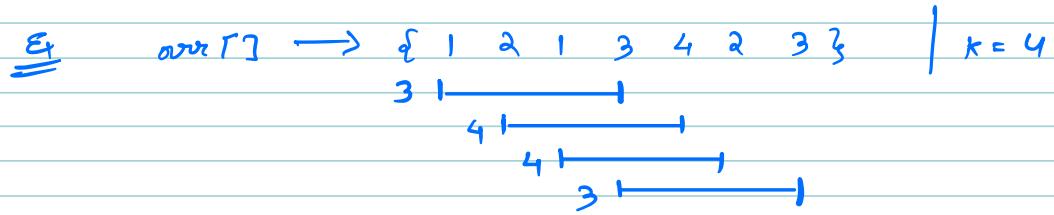
We need to handle when $i == 0$

① check if $PF == K$ at any time.

② Add 0 in the set before iterating

TC → O(N)
SC → O(N)

Question 4 :- Given an array of Integers & a number k . Find the count of distinct elements in every window of size k in the array.



Brute force

↪ Make set for all the windows & print set.size().

Q what will be start of last window?

start	end	size
-------	-----	------

x	$(n-1)$	k
-----	---------	-----

$$\begin{aligned} \Rightarrow n - x - x + k &= k \\ \Rightarrow n - x &= k \\ \Rightarrow x &= n - k \end{aligned}$$

$$[\text{tab}] = b - a + 1$$

PSEUDO CODE

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

```
    HashSet<Int> set;
```

```
    for (j = 0; j < k; j++) {
```

```
        set.add(A[j+i]);
```

```
    }
```

$$TC \rightarrow O((N-k+1) * k)$$

when $k = \frac{N}{2}$ (worst case)

$$\rightarrow O\left(\left(N - \frac{N}{2} + 1\right) * \frac{N}{2}\right)$$

$$\rightarrow O\left(\left(\frac{N}{2} + 1\right) * \frac{N}{2}\right)$$

$$\approx O(N^2)$$

$$SC \rightarrow O(k)$$

> OPTIMIZATION

arr[] $\rightarrow \{1, 2, 1, 3, 4, 3, 3\} \quad | \quad k = 4$

first window :- 1 2 1 3
old ↑ new ↓

second window :- 2 1 3 4
old ↑ new ↓

Third window :- 1 3 4 3
new ↓ old ↑

Hashmap
1 → 21
2 → 10
3 → 12
4 → 1

For
Second
Window

Steps :- ① Create map for first window.

② Add / ^{increment freq of} New element in Next window &
remove / Decrement freq of old element.

③ If Decrement freq becomes 0 for old
element , then remove it from map.

PSEUDO CODE

```
HashMap <int , int> hm;  
for (i=0 ; i<k ; i++) {  
    if ( hm. search ( A[i] ) )  
    |   hm [ A[i] ] += 1;  
    } else {  
    |   hm [ A[i] ] = 1;  
    }  
    Point ( hm. size() );  
    s = 1;  
    e = k;  
    while ( e < n ) {  
        hm [ A[s-1] ] -= 1;  
        if ( hm [ A[s-1] ] == 0 ) {  
        |   hm. remove ( A[s-1] );  
        // Add new element  
        if ( hm. search ( A[e] ) )  
        |   hm [ A[e] ] += 1;  
        } else {  
        |   hm [ A[e] ] = 1;  
        }  
        Point ( hm. size() );  
        s++;  
        e++;  
    }  
}
```

$TC \rightarrow OCN$

$SC \rightarrow OCK$