

Hashing 2 Problems

AGENDA:

Rajat Sharma	_____	Paire Sum k
Sarthak		
Naval Oli	_____	# Paire sum k
Divyanshu		
Saurabh Ruikar	_____	subarray sum k
Khushi Raj	_____	Distinct elements in window k
Shrikanth		
sharath r		
Subhranil Kundu		
Vishal Mosa		
amit khandelwal		
Murali Mudigonda		
soumya hubballi		
Vasanth		
Pankaj		
Rajendra		
SHIVAM PANDEY		
Shivansh Agarwal		
shilpa mamillapalli		
Madhan Kumar M S		
Akansh Nirmal		
Sumit Adwani		
PREETHAM		
Sushant Patil		
thulasi babu		
Sneha L		
Murali krishna Talluri		
Rajeeb Kumar Nath		
Divya P		
Nikhil Pandey		
suyash gupta		
Gowtham Sankaran		
Bhaveshkumar		
Vimal		
Sahithi Kurmachalam		
Sanket Giri		
Gagan Kumar S		
Purusharth A		
Shani		
Shradha Srivastava		
Abhishek Sharma		
Sridhar Hissaria		

Avg Psp

65%



70%

Pair Sum K

Given $A[N]$ and K . Check if there exists a pair (i, j) such that

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

$A =$

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

$K =$
6
22
8

$$A[2] + A[5] = 1 + 5 = 6$$

no pair

$$A[4] + A[8] = 4 + 4 = 8$$

and
true
false
true

Bruteforce

```
for i → 0 to N-1 {  
  for j → 0 to N-1 {  
    if (A[i] + A[j] == K and i != j) {  
      return true  
    }  
  }  
}  
return false
```

$j \rightarrow i+1 \text{ to } N-1$

TC: $O(N^2)$

SC: $O(1)$

QUIZ

$$A = 3 \ 5 \ 1 \ 2 \ 1 \ 2$$

$$k = 7$$

ans = true

$$A = 3 \ 5 \ 1 \ 2 \ 1 \ 2$$

$$k = 10$$

ans = false

Approach \rightarrow use Hashset

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

$$a + b = k$$

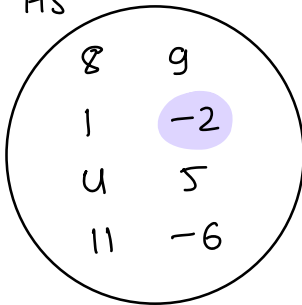
$$b = k - a$$

If a is current value $b = k - a$

	0	1	2	3	4	5	6	7	8
A =	8	9	1	-2	4	5	11	-6	4
b = 6 - a	-2								

HS

$$k = 6$$

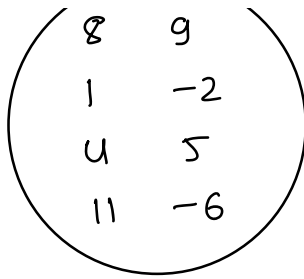


	0	1	2	3	4	5	6	7	8
A =	8	9	1	-2	4	5	11	-6	4
b = 18 - a	10	9							

HS

$$k = 18$$





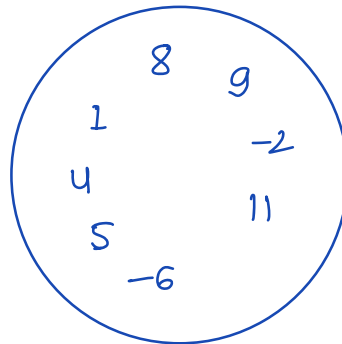
↘ This idea fails for $k=18$
 since we get $A[i] + A[j] = 18$
 but $i \neq j$

check if value $k - A[i]$ is present in range $[0, i-1]$

	0	1	2	3	4	5	6	7	8
A =	8	9	1	-2	4	5	11	-6	4
b = 22 - a	14	13	21	24	18	17	11	28	18

$$k = 22$$

HS



Pseudocode

hs // HashSet

TC: $O(N)$

SC: $O(N)$

```

for (i → 0 to N-1) {
    b = k - A[i]
    if (hs.contains(b)) { return true }
    hs.add(A[i])
}
return false
    
```

Dry Run

$k = 7$

		✓	✓	✓			
A =		3	5	1	2	1	2
b = k - a		1	-1	3	2	3	2



$k = 18$

		✓	✓	✓	✓	✓	✓	✓	✓
A =		8	9	1	-2	4	5	11	-6
b = 18 - a		10	9	17	20	14	13	7	24

pairs with sum k

Given $A[N]$ and k . Count no. of pairs (i, j) such that

$$A[i] + A[j] == k \text{ and } i \neq j$$

$A =$

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

$k = 10$

$\{0, 4\}$ $\{0, 7\}$ $\{1, 5\}$ $\{1, 3\}$ $\{2, 4\}$
 $\{2, 7\}$ $\{3, 5\}$ $\{4, 6\}$ $\{6, 7\}$

$A =$

	3	5	1	2	1	2
$b = 3 - A[i]$	0	-2	2	1	2	1

$k = 3$

ans

	0	0	0	1	2	4
--	---	---	---	---	---	---

We need HashMap since our ans gets incremented by freq. of b everytime.

Pseudocode

hm // Hash Map

ans = 0

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

TC: $O(N)$

SC: $O(N)$

```

b = k - A[i]
if ( hm.containsKey(b) ) {
    ans += hm.get(b)
}

// search for key in hm
if ( hm.containsKey(A[i]) ) {
    int freq = hm.get(A[i])
    hm.put ( A[i] , freq+1 )
}
else {
    hm.put ( A[i] , 1 )
}
}
return ans

```

	0	1	2	3	4	5	6	7
A =	2	5	2	5	8	5	2	8
b	8	5	8	5	2	5	8	2

Hm

2 : 3 5 : 3 8 : 2

k = 10

ans

+1

+2

+2

+1

+3

= 9

equivalent

code

```

hm.put ( A[i] ,
        hm.getOrDefault(A[i], 0) + 1 ) .

```

Subarray with sum k

Given $A[N]$ check if there exists a subarray with
 $\text{sum} == k$

$A =$

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

$k =$

11	[5, 6]		ans = true					
10	2	3	9	-4	ans = true			
15	-4	1	5	6	2	5	ans = true	

$A =$ 100 20 10

$k = 110$

ans = false

$$\text{sum}(i \rightarrow j) == k$$
$$A[i] + A[i+1] + A[i+2] \dots A[j]$$

$$= \begin{cases} P[j] - P[i-1] == k & \text{if } i > 0 \\ P[j] == k & \text{if } i == 0 \end{cases}$$

$i < j$

→ my current value

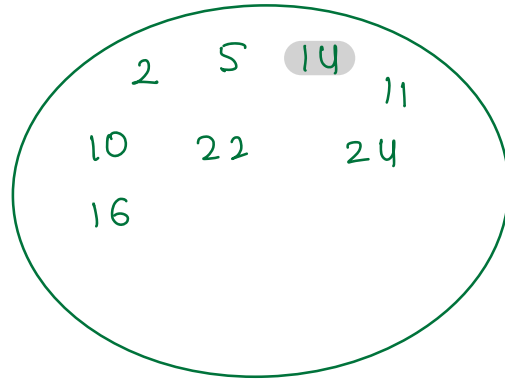
$$a - b = k$$

$$b = a - k$$

$$a == k$$

	0	1	2	3	4	5	6	7	8
A =	2	3	9	-4	1	5	6	2	5
P[] =	2	5	14	10	11	16	22	24	29
b = a - 15	-13	-10	-1	-5	-4	1	7	9	14

HS



k = 15

hs // HashSet

TC: O(N)

total = 0

SC: O(N)

for (i → 0 to N-1) {

total += A[i]

if (total == k) return true

b = total - k

if (hs.contains(b)) { return true }

hs.add(total)

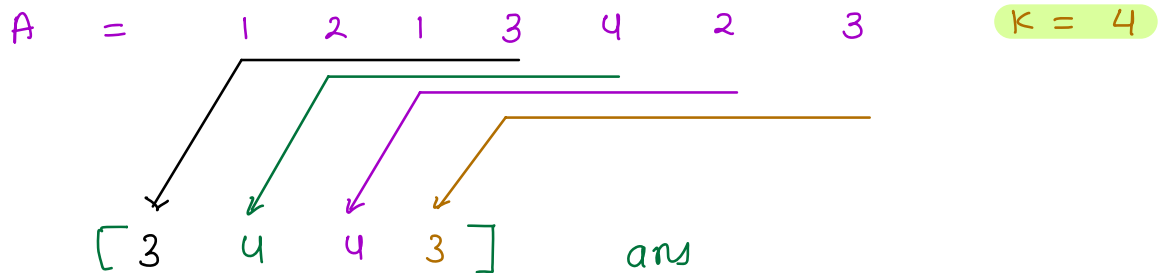
}

return false

Break : 22:58

Distinct elements in every window of size k

Given $A[N]$ and k , Find the count of distinct elements in every window of size k \longrightarrow sliding window



Brute force

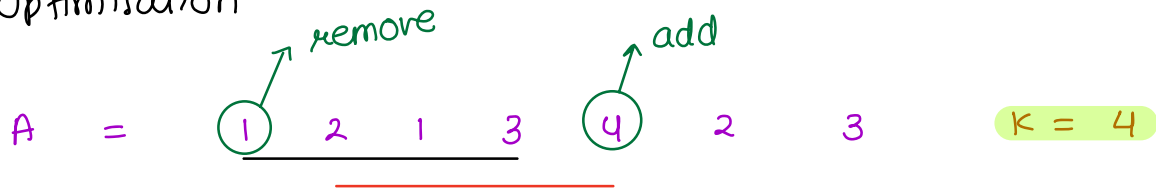
Iterate over all window of size $= k$
and use hashset to count unique elements

```
for i  $\longrightarrow$  0 to  $N-1$  {  
    j = min( $i+k$ ,  $N-1$ )  
    length = j - i + 1  
    if (length  $\neq$  k) continue  
    hs // Hashset  
    for (z  $\longrightarrow$  i to j) {  
        hs.add( $A[z]$ )  
    }  
    print(hs.size())  
}
```

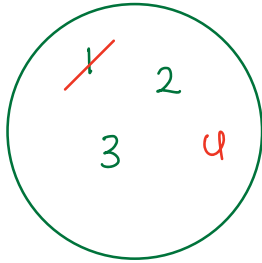
TC: $O(N^2)$

SC: $O(k)$

Optimisation

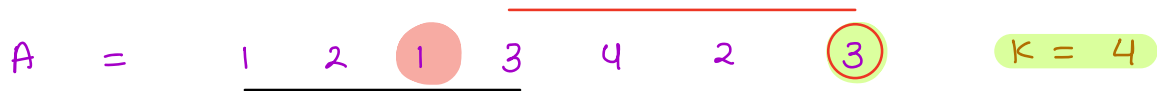


hs

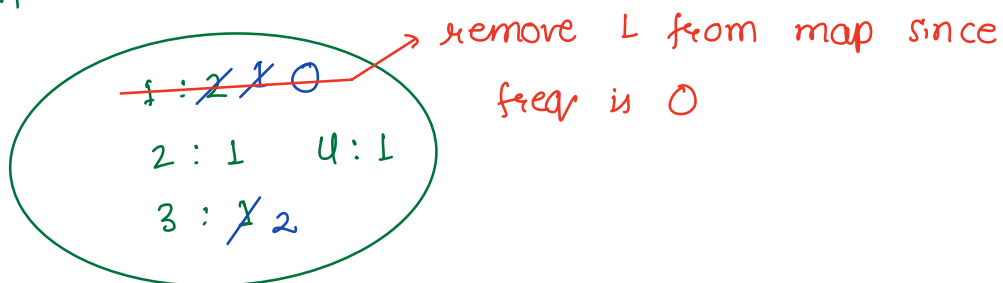


The idea of HS doesn't work since we also need to take into account the frequency of elements.

Using Hash Map



HM



HW

Doubt session