

Q1. Given N array elements, check if there exists a pair (i, j) , such that $arr[i] + arr[j] == k$, ($i \neq j$).

Two sum problem

arr[] \Rightarrow 0 1 2 3 4 5 6 7 8 9
 8 9 1 -2 4 5 11 -6 7 5

	i	j	arr[i]	arr[j]	\Rightarrow	
$k = 11$	4	8	4	7	\Rightarrow	11 ✓
$k = 6$	2	5	1	5	\Rightarrow	6 ✓
$k = 22$	6	6	11	11	\Rightarrow	22 (X) <u>(i != j)</u>

SO^M

check all pairs

for ($i = 0; i < N; i++$) { // a = arr[i]

find b = k - a
in arr

for ($j = i + 1; j < N; j++$) {
 if ($arr[i] + arr[j] == k$) {
 return true
 }
 }

}
 return false

TC $\Rightarrow O(N^2)$
 SC $\Rightarrow O(1)$

try to optimise
 using hashset

$$\underbrace{\text{arr}[i]}_a + \underbrace{\text{arr}[j]}_b = 2k \quad (i \neq j)$$

$$\Rightarrow \underline{\underline{a + b = 2k}}$$

$$\Rightarrow \underline{\underline{b = k - a}}$$

1) create a hashtable

11) populate the hashtable insert all array elements
in hashtable

arr[] \Rightarrow 0 1 2 3 4 5 6 7 8 9
 8 9 1 -2 4 5 11 -6 7 5

$k = 11$

	a	$(k-a)$ b	is b present in hs.
--	---	--------------	---------------------------

8	3	X
---	---	---

9	2	X
---	---	---

1	10	X
---	----	---

-2	13	X
----	----	---

4	7	X
---	---	--------------

↓

hs \Rightarrow 8 9 1 -2 4
5 11 -6 7

↓

return true

$k = \underline{10}$ $\frac{6+5}{15} (9)$ $(i \neq j)$

arr[] \Rightarrow

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

$k = 22$

a	$(k-a)$ b	is b present in hs.
8	14	X
9	13	X
1	21	X
-2	24	X
4	18	X
5	17	X
11	11	

hs \Rightarrow

8	9	1	-2	4
5	11	-6	7	

\checkmark [just checking if $b(k-a)$ is present in hs or not]

* if freq is known this issue can be resolved

1) create hashmap

11) create a freq. hashmap [update him with freq of each element]

arr[] \Rightarrow

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

freq hm \Rightarrow

8:1	1:1	4:1	11:1	-6:1
9:1	-2:1	5:2	7:1	

$k = 22$

a	$(k-a)$ b	is present in arr
8	14	X
9	13	X
1	21	X
-2	24	X
4	18	X
5	17	X
11	11	

if $(a == b)$ {

if (freq(a) > 1)

return true

ex: $\overset{0}{7} \overset{1}{5} \overset{2}{2} \overset{3}{5} \overset{4}{9} \overset{5}{5} \rightarrow \underline{\underline{10}}$

freq arr $\left[\begin{array}{ll} 7:1 & 2:1 \\ 5:3 & 9:1 \end{array} \right]$

$k = 10$

a	$(k-a)$ b	present
7	3	X
5	5	

if $(a == b) \Rightarrow$ return

if (freq(a) > 1) return true

pseudo

1) create a map
and update with frequencies.] $O(N)$

$O(N)$ [

```
for (i=0; i<N; i++) {  
    a = arr[i]  
    b = K - a  
  
    if (a != b) {  
        if (b is present in map)  
            return true  
    } else {  
        if (freq(a) > 1)  
            return true  
    }  
}  
  
return false
```

$T.C \approx O(2N) \approx O(N)$
$S.C \approx O(N)$

⇒ solving with Hashset :-

arr ⇒

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

hs ⇒ $[8, 9, 1, -2, 4]$

⇒ when iterating to insert in hs, when at i^{th} index we have all elements from $[0 - (i-1)]$ in hs.

arr ⇒

0	1	2	3	4	5	6	7	8	9
8	9	1	-2	4	5	<u>11</u>	-6	7	5

k = 22	a	b (k-a)	HS.	Insert-
	8	14(x)	[]	8
	9	13(x)	[8]	9
	1	21(x)	[8, 9]	1
	-2	24(x)	[8, 9, 1]	-2
	4	18(x)	[8, 9, 1, -2]	4
	5	17(x)	[8, 9, 1, -2, 4]	5
	11	<u>11(x)</u>	[8, 9, 1, -2, 4, 5]	11
			[8, 9, 1, -2, 4, 5, 11]	

pseudo

```

hashset<int> hs
for (i = 0; i < N; i++) {
    a = arr[i]
    b = k - a
    if (b is present in hs)
        return true
    else {
        hs.insert(a)
    }
}
return false

```

$TC \Rightarrow O(N)$
 $SC \Rightarrow O(N)$

\Rightarrow

0	1	2	3	4	5
7	5	2	5	9	5

k = 10	a	b	HS	Present
	7	3	[]	f
	5	5(x)	[7]	T
	2	8(x)	[7, 5]	T
	5	5(✓)	[7, 5, 2]	T
		<u>5</u>	<u>5</u>	
		↓	True	

How

1) Calculate no. of (i, j) pairs such that—

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

Hint \rightarrow hashmap only

2) Calculate if there exists a pair (i, j) , such that

$$\text{arr}[i] - \text{arr}[j] = k, \quad \underline{\underline{\{i \neq j\}}}$$

Q. 2. Given N array elements, calculate no. of distinct elements in every window of size ' k '.

arr[] \rightarrow 0 1 2 3 4 5 6 7 8 9
 2 4 3 8 3 9 4 9 4 10

$$k = 4$$

Subarray

distinct

[0-3]

\Rightarrow

4

[3-6]

\Rightarrow

4

[1-4]

\Rightarrow

3

[4-7]

\Rightarrow

3

[2-5]

\Rightarrow

3

\downarrow o/p

[5-8]

\Rightarrow

2

[6-9]

\Rightarrow

3

[7-10]

o/p

Prob 1 one

for every subarray of size k ,

get no. of distinct element using hashmap

2) How many subarrays of len k are present
in array of len N . $\Rightarrow [N-k+1]$

* no. of subarrays $\Rightarrow (N-k+1)$

* TC for each subarray $\Rightarrow O(k)$

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

if $k \leq N/2$

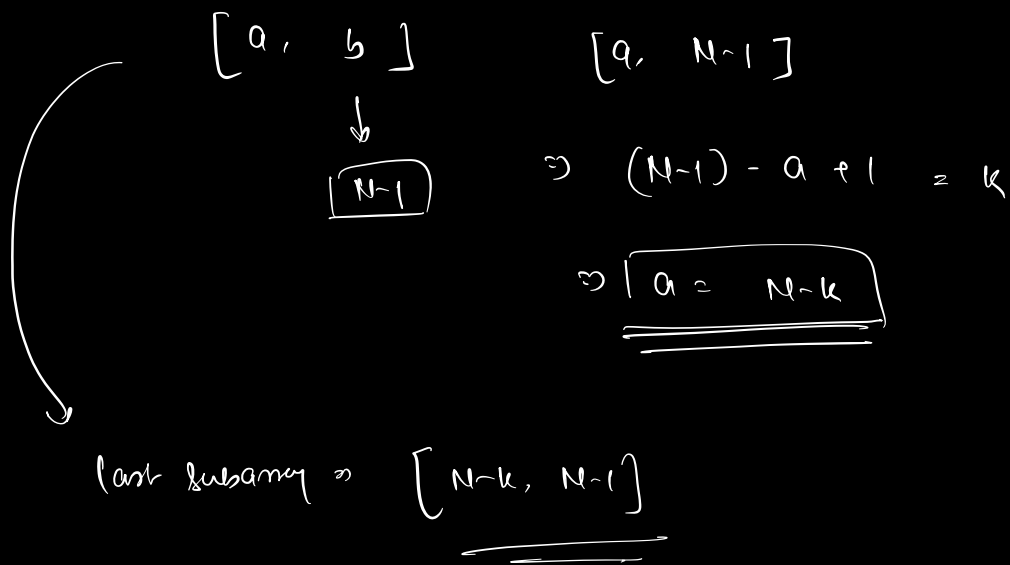
$$\hookrightarrow O(N^2)$$

$$SC \Rightarrow \underline{O(k)}$$

len of subarray $\Rightarrow k$,

what would be last subarray indices

$$\left\{ \underline{[a \ b] = b - a + 1} \right\}$$



pseudo

HashSet<int> hs;

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

hs = new HashSet<>

← creating a hs object
for each
subarray

for (j=i; j < (i+k); j++) {

hs.insert(arr[j]);

}

print (hs.size());

}

$TC \Rightarrow O(N^2)$
 $SC \Rightarrow O(k)$

⇒ Sliding window with Hashset

arr ⇒ 2 4 3 8 3 9 4 9 4 10, k=4

	Subarray	remove	add	hs	distinct
1st	[0 3]			{2, 4, 3, 8} X	4
	[1 4]	0 th (2)	4 th (3)	{4, 3, 8} X	3
	[2 5]	1 st (4)	5 th (9)	{3, 8, 9} X ↑	3
	[3 6]	2 nd → (3)	6 th (4)	{8, 9, 4} 3 ↑	

XX logic doesn't work

Hashmap with sliding window

arr[] ⇒ 2 4 3 8 3 9 4 9 4 10

sub	remove	add	HM	distinct
[0-3]			[2:1, 3:1, 4:1, 8:1] 8:1	4
[1-4]	0 th [2]	4 th [3]	[2:0, 3:2, 4:1, 8:1]	3

[2-5]	1 st [4]	5 th [9]	[3:2, 4:0, 8:1, 9:1]	2
[3-6]	2 nd [3]	6 th [4]	[3:1, 4:1, 8:1, 9:1]	4
[4-7]	3 rd [8]	7 th [9]	[3:1, 4:1, 8:0, 9:2]	3
[5-8]				
[6-9]				

flow

i) Create a hm

ii) store freq for first subarray of len k.

iii) reduce freq of $arr[i]$ being removed

increase freq if $arr[i]$ is added

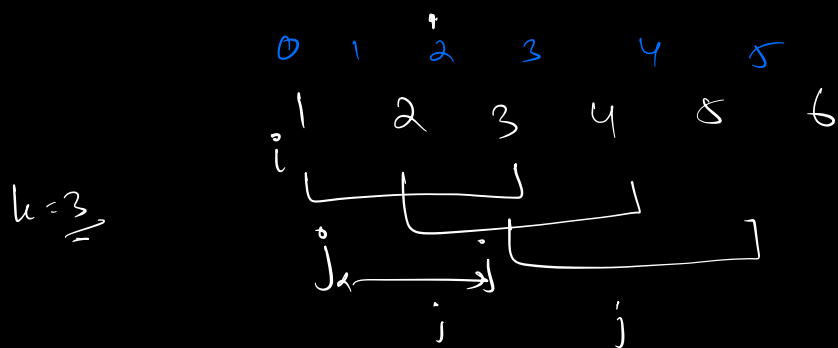
or, if not already present

$\langle arr[i], 1 \rangle$

iv) if $freq = 0$, remove the element

POD
code?

$$\begin{aligned} TC &\Rightarrow O(N) \\ SC &\Rightarrow O(k) \end{aligned}$$



$$\underline{j=i} : j \leq \underline{(i+k)}$$

$$\underline{0+3} = 3$$