Q1) Given N array elements, check if there exists a pair (i,j) such that ar[i] + ar[j] = K && i != j

index values

a + b = K

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

K = 11 i = 4 j = 8

K = 6 i = 0 j = 3

K = 22 i = X j = X

Brute force: Nested loops

TC: $O(n^2)$

$a + b = K \implies b = K - a$

K = 11 a = 5 b = 6

a	b (K-a)
8	3
9	2
1	10
-2	13
4	7 ✓

K = 11

Hashset

Allows me to check if b exists or not.

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

K=22

HS = { 8, 9, 1, -2, 4, 5, 11, -6, 7 }

a	b(k-a)
8	14
9	13
1	21
-2	24
4	18
5	17
11	11

K = 22

~~Yes~~

Learning: We need to maintain freq.

Freq Hashmap

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

K=22

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

a	b	
8	14	X
9	13	X
1	21	X
-2	24	X
4	28	X
5	12	X
11	11	} freq(11) > 1
	X	

-6	28	X
7	15	X
5	12	X

return false

Pseudo Code

1) Create the frequency hm.

```
for(i=0; i<n; i++) {
    a = ar[i]      b = k-a
    if (a == b) {
        if ( hm.get(a) >= 2 )
            return true.
    }
    else {
        if ( hm.containskey(b) )
            return true.
    }
}
return false
```

TC: O(N) SC: O(N)

$$a + b = k$$

TODO: 1) $a - b = k$

$a = k + b$

$b = a - k$

OA

Q2 Calc the number of distinct elements in each subarray of size k.

Morgan Stanley

	0	1	2	3	4	5	6	7	8	9
Eg: arr[10] =	2	4	3	8	3	9	4	9	4	10

k = 4

Idea: Hashset

0, 3	2 4 3 8	{4, 3, 8}	⇒ 4 ✓
1, 4	4 3 8 3		⇒ 3 ✓
2, 5	3 8 3 9		⇒ 3 ✓
3, 6	8 3 9 4		⇒ 4 ✗
4, 7			
5, 8			
6, 9			

Saviour Freq hm

Idea: Optimize using hashmap

0 : 3
1 : 4

<2,1> <4,1> <3,1> <8,1>

add ⇒ freq ++
sub ⇒ freq -- but if freq = 0 remove

<1,4> remove q_0 add a_4 <4,1> <3,2> <8,1>

<2,5> remove q_1 add a_5 <9,1> <3,2> <8,1>

<3,6> remove q_2 add a_6 <9,1> <3,1> <4,1> <8,1>

Code

```
hashmap<int, int> hm
for (i=0 ; i<k ; i++) {
    hm[ a[i] ]++
}
print (hm.size())
s=1        e=k
while (e<n) {
    hm[ar[s-1]]--
    if ( hm[ar[s-1]] ==0 )
        hm.remove ( ar[s-1] )

    hm[ar[e]]++
    print (hm.size)
    s++                e++
}
```

O(n)

O(n)

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

⇒ O(N)

Done

$[0:i) \Rightarrow 0$

$0 \Rightarrow 1$