Todays content

1) Pair Sum = k

2) Distinct elements in every window of size = k

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

Morgan Stanley, GS

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

(a = index 6, b = index 8)

K=11 i=4 j=8

K=6 i=2 j=5

K=22 i=6 j=6 ×

no such pair

Brute force: 2 nested loops

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

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

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

a = b

Learning: we need frequency of each elem

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
9	13
1	21
-2	24
4	18
5	17
11	11
-6	28
7	15
5	17

if a = b, freq > 1

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) > 1 )
            return true.
    }
    else {
        if ( hm.containskey(b) )
            return true.
    }
}
return false
```

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

$a + b = k$

TODO:

1) $a - b = k$ $\quad b = a - k$

2) $a + b = k$ check if it exists.

No of such pairs.

count

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

Adobe

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

k=4 k<n

0, 3	2 4 3 8	⇒4	2 4 3 8
1, 4	4 3 8 3	⇒3	4 3 8
2, 5	3 8 3 9	⇒3	9 3 8
3, 6	8 3 9 4	⇒4	9 4 8
4, 7			
5, 8			
6, 9			

Idea: Hashset

Saviour : Hashmap

Idea: Optimize using hashmap

0:3 <2,1> <4,1> <3,1> <8,1>

insert ⇒ freq ++

remove ⇒ freq --

if freq = 0 remove the entry

<1,4> insert a[4] ⇒ 3 <4,1> <3,2>
s e remove a[0] ⇒ 2 <8,1>

<2,5> insert a[5] ⇒ 9 <9,1> <3,2>
s e remove a[1] ⇒ 4 <8,1>

<3,6> insert a[6] ⇒ 4 <9,1> <3,1>
s e remove a[2] ⇒ 3 <4,1> <8,1>

Code

0, k-1

```
hashmap<int, int> hm          1, k
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++
}
```

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

{done}