7:05AM

Topics: - arrays intro
- P1 # of elements have at least a larger num.
- P2 - a[i]+a[j]=k
- P3 reverse array
- P4 reverse array partial
- P5 rotate array k times.

300

200+ Coding Problem

10 min

30 min

break

Array is: a collection of data sequential in memory, same type

ex int a[10] a[i] a+i → 1000 + 4 × i = 1016 (int)

Quiz: rang of index for array size n [0, n-1]

Quiz: accessing any element in array is O(1)

- How to print an array?

```
void printArr(int a[]){
    n = a.Len
    for(i=0; i<n; i++){
        print(a[i])  → O(1)
    }
}
```

Quiz
TC: O(n)
SC: O(1)

🚨🚫Answer or solution in private chat to instructor only,
Question in public chat

P1 Given an array of `n` integers, `count` (int) the number of elements, having at least one element greater than itself.

ex as{-3, 2, 6, 8, 4, 8, 5}
0 1 2 3 4 5 6 ← index

count +1 +1 +1 +1 +1 ⇒ ans = 5

ex as{2, 3, 10, 7, 3, 2, 10, 8}
0 1 2 3 4 5 6 7 ← index

count +1 +1 +1 +1 +1 +1 ⇒ ans = 6

Quiz as{2, 5, 1, 4, 8, 0, 8, 1, 3, 8} ans = 7

ex as{8, 8, 8, 8, 8} ans = 0

① find max element

② Count those element that are not max
< max

(2') Count freq of max → ret a.Len - freqMax

Psudo Code

Quiz

TC: O(n)

SC: O(1)

```
int CountMax(int arr[]){
    n=arr.len
    maxE = a[0]          INT_MIN
    for(i=0; i<n; i++){      1
        if( arr[i] > maxE){
            maxE = arr[i]
        }
    }
    Count = 0
    for(i=0; i<n; i++){
        if(arr[i] != maxE)  Count += 1
    }
    ret Count
}
```

step1

step2

O(n)

+

O(n)

O(n+n) = O(2n)

arr[i] == maxE

ret n - Count

Google

true / false

P2 Given an array (int) with n elements, check if there exists a pair {i, j} such that a[i] + a[j] == k & i ≠ j

(i, j: index)

ex as{3, -2, 1, 4, 3, 6, 8} ans = true
0 1 2 3 4 5 6
k = 10

as{3, 5, 2, 1, -3, 7, 8, 15, 6, 13} ans = true
k = 10 ans = 3

ex as{2, 4, -3, 7} ans = false
0 1 2 3
k = 5

ex as{2, 4, -3, 7} ans = false
0 1 2 3
k = 8

ex as{2, 3, 7, -4, 3} ans = true
0 1 2 3 4
k = 6

idea 1

```
bool checkPair1(int a[], k){
    n = a.Len
    for (i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(i != j && a[i]+a[j]==k ){
                ret true
            }
        }
    }
    ret false
}
```

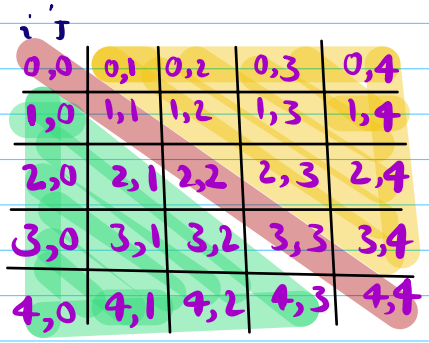
Quiz

TC: $O(n^2)$

SC: $O(1)$

a[i]+a[j] = a[j]+a[i]

2, 4, -3, 7, 8
0 1 2 3 4



optimized

idea 2

Quiz

```
bool checkPair2(int a[], k){
    n = a.Len
    for(i=0; i<n-1; i++){
        for(j=i+1; j<n; j++){
            if(i!=j && a[i]+a[j]==k){
                ret true
            }
        }
    }
    ret false
}
```

i<j → upper triangle

not needed anymore

i	j	itr
0	[1, n-1]	n-1
1	[2, n-1]	n-2
2	[3, n-1]	n-3
⋮	⋮	⋮
n-2	[n-1, n-1]	1

\# iters

$S = 1+2+3+\dots+n-1$

$= \frac{(n-1)((n-1)+1)}{2} = \frac{(n-1)n}{2}$

$S = \frac{N(N+1)}{2}$ $N = n-1$

$1+2+3+\dots+N$

$\frac{(n-1)n}{2} = \frac{n^2}{2} - n$

$= O(n^2)$ TC

Quiz ans

SC:

How to swap two variables?

int a, int b

temp

```
int temp = a
a = b
b = temp
```

before		after	
a	b	a	b
3	5	5	3

temp: 3

P3 Given an int array reverse the entire array.

Constraint → SC: O(1)

ex as { 1, 2, 3, 4, 5 }

{ 5, 4, 3, 2, 1 }

original array

10	4	7	6	-2	17	8	-1

ex as { -1 | 8 | 17 | -2 | 6 | 7 | 4 | 10 }

0 1 2 3 4 5 6 7

++i
i++
a[i]

i++ j--

swap

i		J
0	<	7
1	<	6
2	<	5
3	<	4
4	>	3

as long as i < J

ex as { -1 | 6 | 3 | 2 | 8 | 9 | 10 }

0 1 2 3 4 5 6

i		J
0	<	6
1	<	5
2	<	4
3	==	3
4	>	2

Psudo code

```
void reverseArray(int a[])
    n = a.Len
    i = 0      j = n-1
    while(i<j){ → it n/2
        int temp = a[i]
        a[i] = a[j]
        a[j] = temp
        i++
        j--
    }
}
```

swap 🔄 O(3) = O(1)

Quiz

O(n) ← TC

O(1) ← SC

P4 Given an array (int, n element) and s_i & e_i ($s_i <= e_i$, $e_i < n$)

start index, end index

reverse the array from index s_i to e_i.

Constraint → SC: O(1)

ex a[]: { -3 | 4 | 2 | 8 | 7 | 9 | 6 | 2 | 10 } $s_i = 3$, $e_i = 7$

0 1 2 3 4 5 6 7 8

-3 | 4 | 2 | 2 | 6 | 9 | 7 | 8 | 10

reversed

Code:

```
void reverseArray2(int a[], int si, int ei)
    n = a.Len
    i = si   j = ei        ✓
    while(i < j){
        int temp = a[i]
        a[i] = a[j]
        a[j] = temp
        i++
        j--
    }
}
```

O(n) ← TC

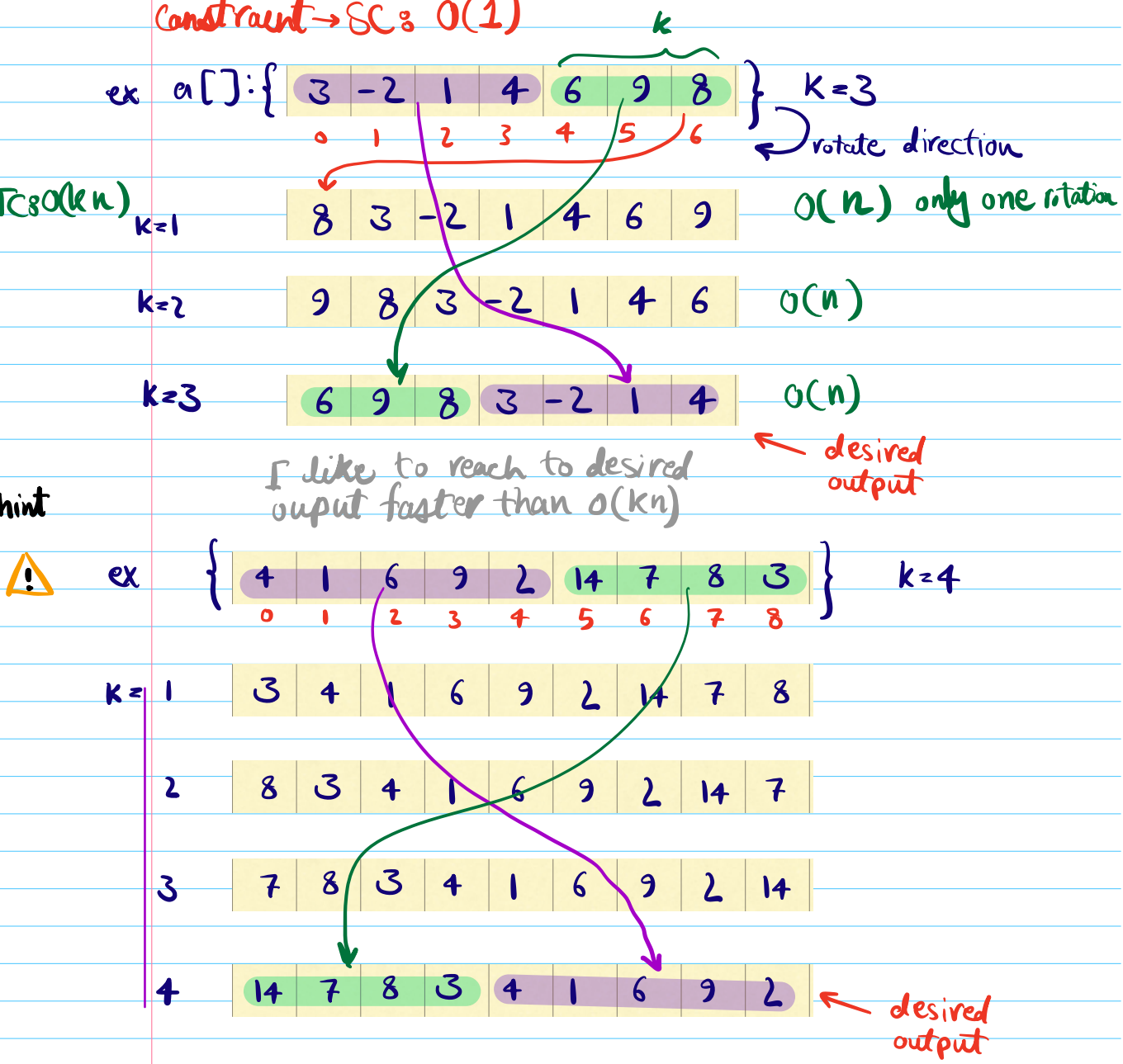
O(1) ← SC

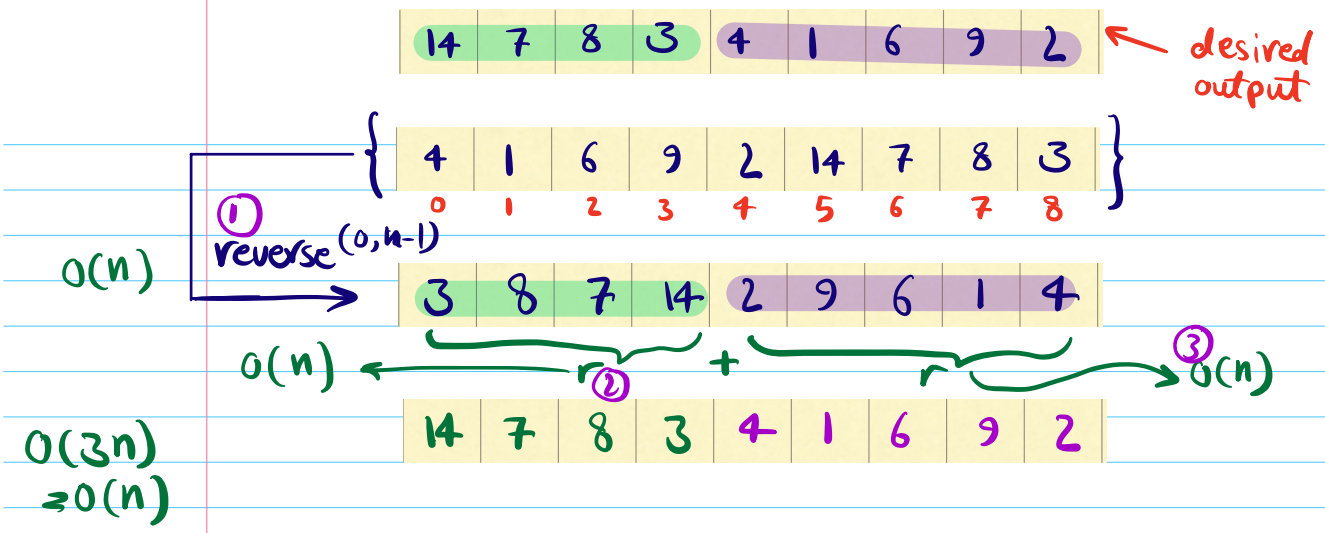
2017

Facebook

P5 Given an array (n elements) rotate array from last to first by k times

Constraint → SC: O(1)





O(3n)
=O(n)

Code:

```
void kRotate(a[], k){
    k = k%n
    ①reverseArray2(a, 0, n-1)
    ②reverseArray2(a, 0, k-1)
    ③reverseArray2(a, k, n-1)
}
```

TC:

SC:

Large ks: {1,2,3,4} k=4

k=1 4,1,2,3

k=2 3,4,1,2

k=3 2,3,4,1

k=4 1,2,3,4

k = k%n

k = k%n = 0

0 → n

= k%n

5%4 = 1