

Intro to arrays

int arr[5] // array of integers  
of size 5

First element  $\rightarrow$  arr[0]

0 1 2 - - - - - n-1  
 $i \in [0, n-1]$

arr[i]  $\rightarrow O(1)$  } why?  
arr[0]  $\rightarrow O(1)$  } linked List  
arr[n-1]  $\rightarrow O(1)$  class

# Print all numbers of array

```
for (i=0; i<N; i++)  
    print (arr[i])
```

y

TC:  $O(N)$

SC:  $O(1)$

Q1 Given integer array of N elements.  
Count number of elements having  
atleast 1 number greater than  
itself.

Eg - {-3, -2, 6, 8, 4, 8, 5}  
ans = 5

Obs : 1) All the max numbers are  
not counted  
2) Everything other than max  
will be counted.

$$\text{Ans} = N - \text{count\_of\_max}$$

- 1) Find the max
- 2) Find the count of max
- 3) Return  $N - \text{count}$

1) `int max_val = INT_MIN // -∞`  
minimum value for  
int data type

`for (i=0; i < N; i++)`  
  `{`  
    `if ( arr(i) > max_val )`  
      `max_val = arr(i)`  
  `}`

$\left\{ \begin{array}{r} 1, 2, 3, 1, 5 \\ 0, 1, 2, 3, 4 \end{array} \right\} \rightarrow -\infty$

i	max_val
0	1
1	2
2	3
3	3
4	5

2) Count occurrence of max-val

```
int count=0  
for ( i=0 ; i<N ; i++ )  
{  
    if ( arr(i) == max_val )  
        count++  
}
```

3) return  $N - \text{count}$

$$TC: N + N = 2N = O(N)$$

$$SC: O(1)$$

Q2 Given N array elements, check if there is a pair  $i, j$  such that  $\text{arr}[i] + \text{arr}[j] = K$  and  $i \neq j$ .

$i$  &  $j$  are indexes.  $K$  is given sum.

Eg  $\text{arr} [0, 1, 2, 3, 4, 5, 6] = [3, -2, 1, 4, 3, 6, 8]$

$$K = 10$$

$$\text{arr}[3] + \text{arr}[5] = 10$$

Eg -  $\text{arr} [0, 1, 2, 3, 4, 5, 6] = [2, 4, -3, 7]$   $K=8$

ans = false

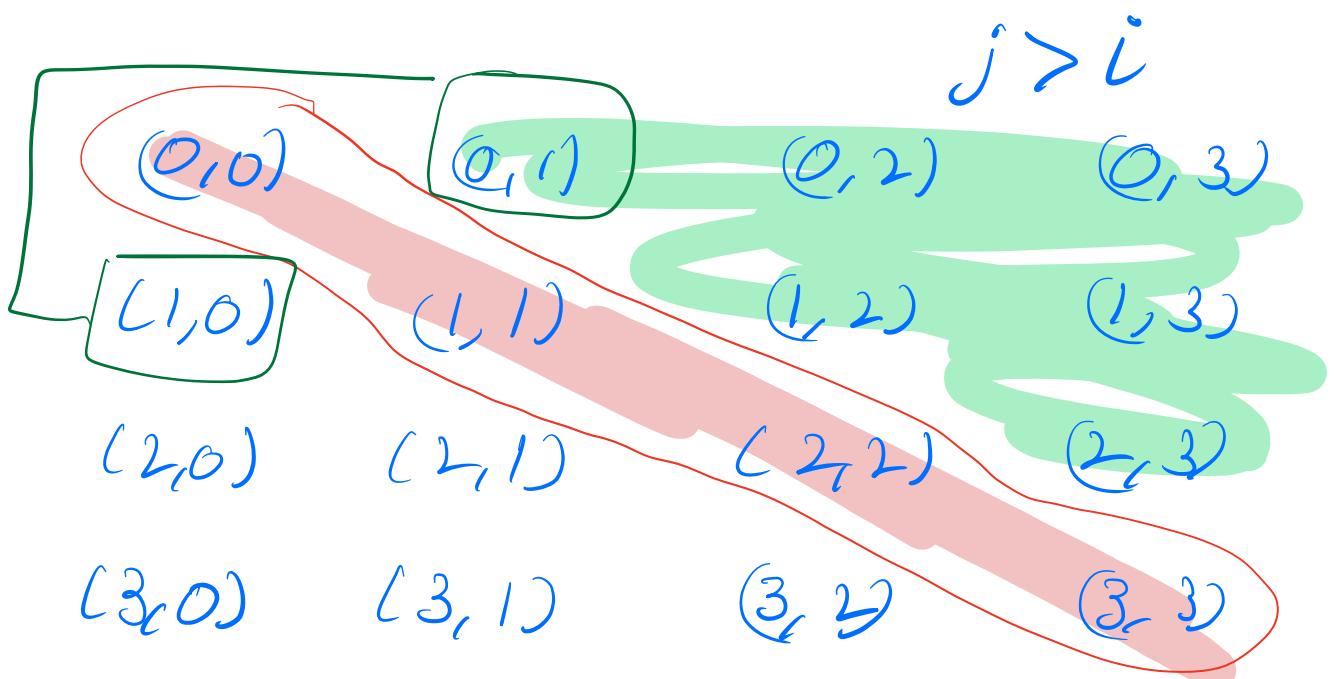
~~for (i=0 ; i<N ; i++)~~

~~for (j=0 ; j<N ; j++)~~

~~if (arr[i]+arr[j] == K &&  
        i != j)  
      return true~~

y  
y

return false



$(0,1)$   $\text{arr}[0] + \text{arr}[1]$   
 $(1,0)$   $\text{arr}[1] + \text{arr}[0]$

$3+7$   
 $7+3$

for ( $i=0$ ;  $i < N$ ;  $i++$ )

  for ( $j=i+1$ ;  $j < N$ ;  $j++$ )

    if ( $\text{arr}(i) + \text{arr}(j) == K$    &  
           $i \neq j$ )  
      return true

y

Number of iterations,

i	j	$b-a+1$
0	$[1, N-1]$	$N-1$
1	$[2, N-1]$	$N-2$
2		$N-3$
⋮		$N-4$
		⋮

$N-2$   
 $N-1$

$[N-1, N-1]$

1  
0

$$\text{Total} = 0 + 1 + 2 + \dots + N-1$$

Sum of  $N$  natural no.s

$$= \frac{N(N+1)}{2} \quad \text{Put } N=N-1$$

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

TC:  $O(N^2)$

SC:  $O(1)$

$\mathcal{L}$       1, 2, 3,  $\mathcal{Y}$        $K=5$

$$i=0$$

$$i=0$$

$$i=1$$

$$j=1$$

$$j=2$$

$$j=2$$

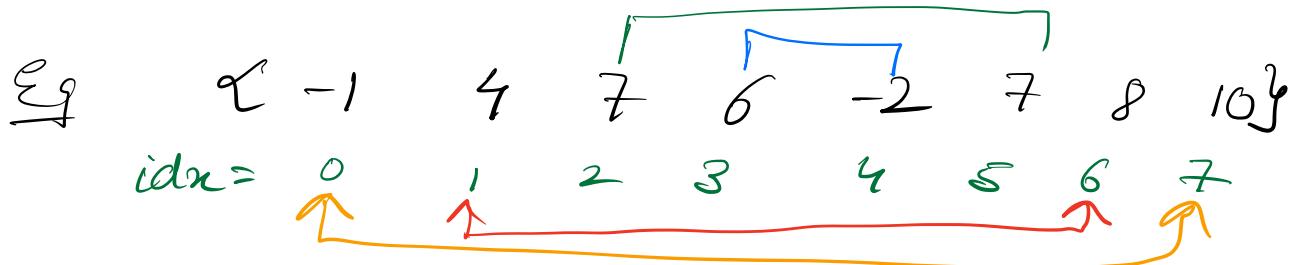
$$s=3$$

$$s=4$$

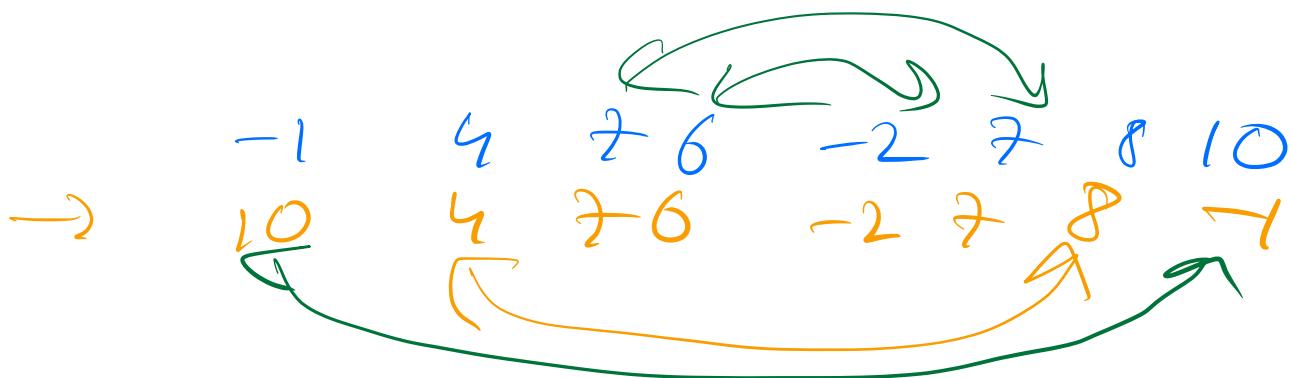
$$s=5$$

return  
true.

Q3 Given an integer array, Reverse the entire array. [ O(1) SC ]



$\rightarrow \{ 10, 8, 7, -2, 6, 7, 4, -1 \}$



i	j	
0	7	swap
1	6	swap
2	5	swap
3	4	swap
4	3	STOP

$N=7$

$\{1, 2, 3, 4, 5\}$

$i$	$j$		
0	4		swap
1	3	$n=5$	swap
2	2		STOP

STOP when  $i \geq j$

Code reverse (int arr[], int  $N$ )

X      int  $i = 0$   
      int  $j = n - 1$

while ( $i < j$ )

X      swap(arr[i], arr[j])  
                 $i++$   
                 $j--$

Y      TC:  $O(N)$       SC:  $O(1)$

O Reverse the part of  
array starting at index s  
ending at index e.

100      200      300      400      500  
0            1            2            3            4

$$s = 1$$
$$e = 3$$

ans =

100      400      300      200      500

reverse part (ints, int arr[], int  $N$ , int  $e$ )

x      int       $i = s$   
      int       $j = e$

while ( $i < j$ )

  x    swap (arr[i], arr[j])  
       $i++$   
       $j--$

y

TC:  $O(N)$

worst case  
when       $s = 0$   
               $e = n-1$

SC:  $O(1)$

O5 Given an array, Rotate the array from last to first  $K$  times.

input

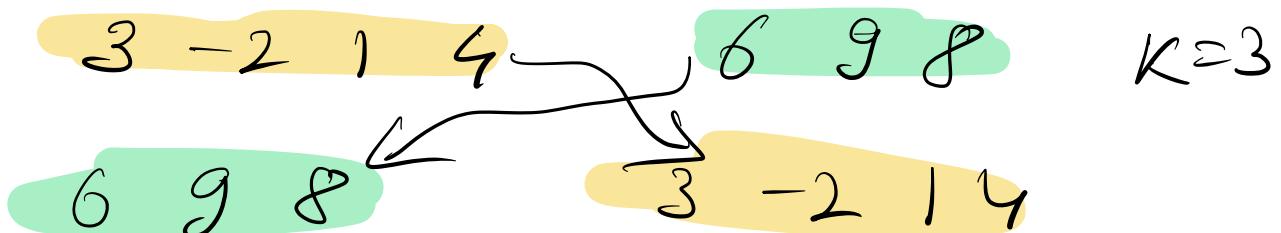
Eg - {3 -2 1 4 6 9 8}

$K=1$  {8 3 -2 1 4 6 9}

$K=2$  {9 8 3 -2 1 4 6}

$K=3$  {6 9 8 3 -2 1 4}

$K=4$  {4 6 9 8 3 -2 1}



Last 3 elements  
⇒ First 3 elements.

1) Reverse the whole array

Array = 3 -2 1 4 6 9 8

8 9 6 4 1 -2 3

6 9 8 3 -2 1 4

2) Reverse first K elements

3) Reverse Last N-K elements

Code

reverse(arr, N)

reverse(arr, N, 0, R-1)

reverse(arr, N, R, n-1)

Imagine  $K > N$

Eg  $\alpha [100, 200]$

$$K = 5$$

11 array out of bound error.

$\alpha [200, 100]$

$K=0$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$K=1$	$a_5$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$
$K=2$	$a_4$	$a_5$	$a_0$	$a_1$	$a_2$	$a_3$
$K=3$	$a_3$	$a_4$	$a_5$	$a_0$	$a_1$	$a_2$
$K=4$	$a_2$	$a_3$	$a_4$	$a_5$	$a_0$	$a_1$
$K=5$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_0$
$K=6$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$K=7$	$a_5$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$
$K=8$	$a_4$	$a_5$	$a_0$	$a_1$	$a_2$	$a_3$
$K=9$	$a_3$	$a_4$	$a_5$	$a_0$	$a_1$	$a_2$

Ans after rotating  $K$  times

= ans after rotating  $K \% N$  times

0, 6, 12, 18 ... have  
 $18 \% 6 = 0$

1, 2, 13, 19 ... have  
 $19 \% 6 = 1$

$$K = K \% N$$

Code

reverse(arr, N)

reverse(arr, N, 0, R-1)

reverse(arr, N, R, n-1)

# Dynamic arrays.

# int arr[5]      size is fixed  
          ^  
static sized array.

Dynamic arrays.

C++	Java	Python
vector	ArrayList	List

vector<int> v

→ C++ syntax

v.push\_back(10) → 10

v.push\_back(20) → {10, 20}

v[i] = arr[i]

~~for (i=0; i< v.size(); i++)  
    print(v[i])~~

Doubts

array size 6

12

6	times	same
12	times	same

↓

5 more times

↓

12

$$12 \div 6 = 2$$

$$9 \div 6 = \underline{3}$$

~~for (i=0; i<5; i++)~~

v - - - -

x int count = 0;  
y