

Arrays

int arr[5] = { 3, 2, 8, 9, 7 }
 0 1 2 3 4

Diagram illustrating array indexing:

- A variable declaration `int arr[N]` is shown.
- An arrow points from the variable to the first element, labeled `1st elem` and `0`.
- An arrow points from the variable to the last element, labeled `last element` and `N-1`.

Quiz [5, -4, 8, 9, 10]
= 0 1 2 3 4

1st element \Rightarrow an [0]

S^{1s} elekt \Rightarrow am [4]

Print all array elements

```
for ( i=0; i < n ; i++ ) {
```

```
    Print ( arr [i] ),
```

```
}
```


O(1)
Constant
time

TC : $O(N)$

Q Given an array of size N . Count the number of elements which have at least one element greater than itself.

$$-3, -2, 6, 8, 4, 8, 5 \Rightarrow 5$$

✓ ✓ ✓ ✗ ✓ ✗ ✓

Qwq 2, 5, 1, 4, 8, 0, 8, ↓ 1, 3, 8 ⇒ 7

Observation

- ① Only the max element will not have a greater elemt.

$$C_{\max} \quad N$$

$$Ans = N - C_{\max}$$

Step I

Find the max no.

Step II

Iterate and find count of max no. (C_{max})

Step III

Return $N - C_{\text{max}}$

// Find the max

and $\text{max} = \emptyset \Rightarrow -\infty / A[0]$,

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

 if ($A[i] > \text{max}$) {

$\text{max} = A[i]$;

 }

}

$A[] : \downarrow -2, \downarrow -8, \downarrow -3$

$C = 0;$

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

 if ($A[i] == \text{max}$) {

$C++$;

 }

return $N - C$;

of iteration = $2N$

TC : $O(N)$ SC : $O(1)$

$\Rightarrow \underline{\text{HW}}$: Implement in a single loop, [30 min]

Q Given an array of size N & a no. K ,

Return True if there exists a pair $a[i], a[j]$

such that

$$a[i] + a[j] = K]$$

$$\& i \neq j$$

A: 3, -2, 1, 4, 3, 5, 6, 8

$$K = 10$$

$$A[3] + A[5] \Rightarrow 10 \checkmark$$

$\Rightarrow T$

$A : \frac{0}{2}, \frac{1}{4}, -\frac{2}{3}, \frac{3}{7}$ $7 + (-3) \neq 5$

$K = 5$

False

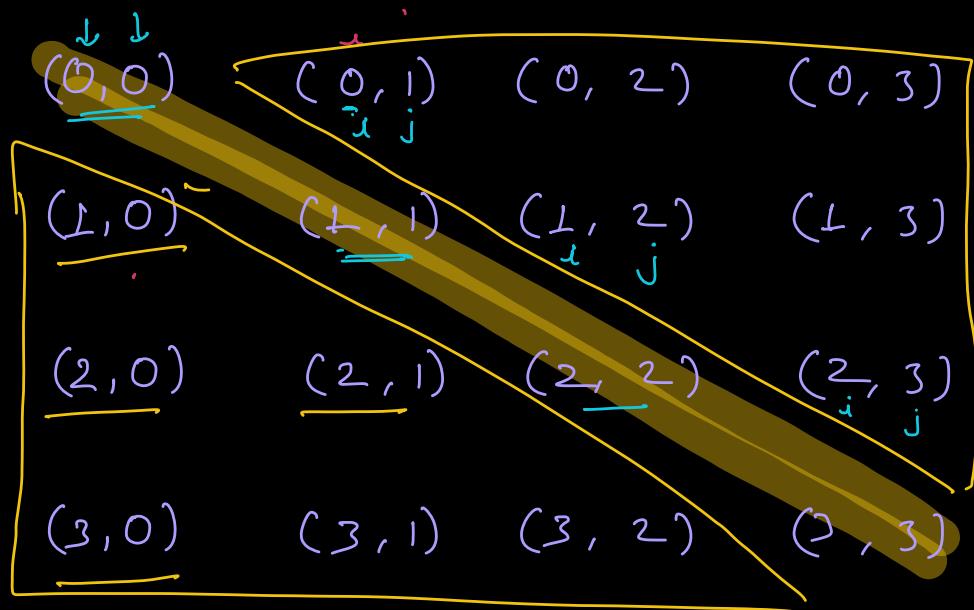
$A : \frac{0}{3}, -\frac{1}{2}, \frac{2}{1}, \frac{3}{4}, \frac{4}{3}, \frac{5}{6}, \frac{6}{8}$



$K = \underline{\underline{10}}$

1) Check for all possible pairs.

$$N = 4 \quad 0, 1, 2, 3$$



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



```

for (i=0; i<N; i++) {
    for (j=0; j<N; j++) {
        if (i != j) {
            if (A[i] + A[j]
                == K)
                ret true;
        }
    }
}

```

ret false;

of iterations = N^2

TC : $O(N^2)$

Upper triangle

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

 for ($j = i+1$; $j < N$; $j++$) {

 if ($a[i] + a[j] == k$) {

ret true;

}

 }

ret false;

$$\# \text{ of iterations} = \frac{N(N-1)}{2}$$

$$TC : O(N^2)$$

i	$j : [i+1, N-1]$	<u>iterations</u>
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
N-1	[N, N-1]	0

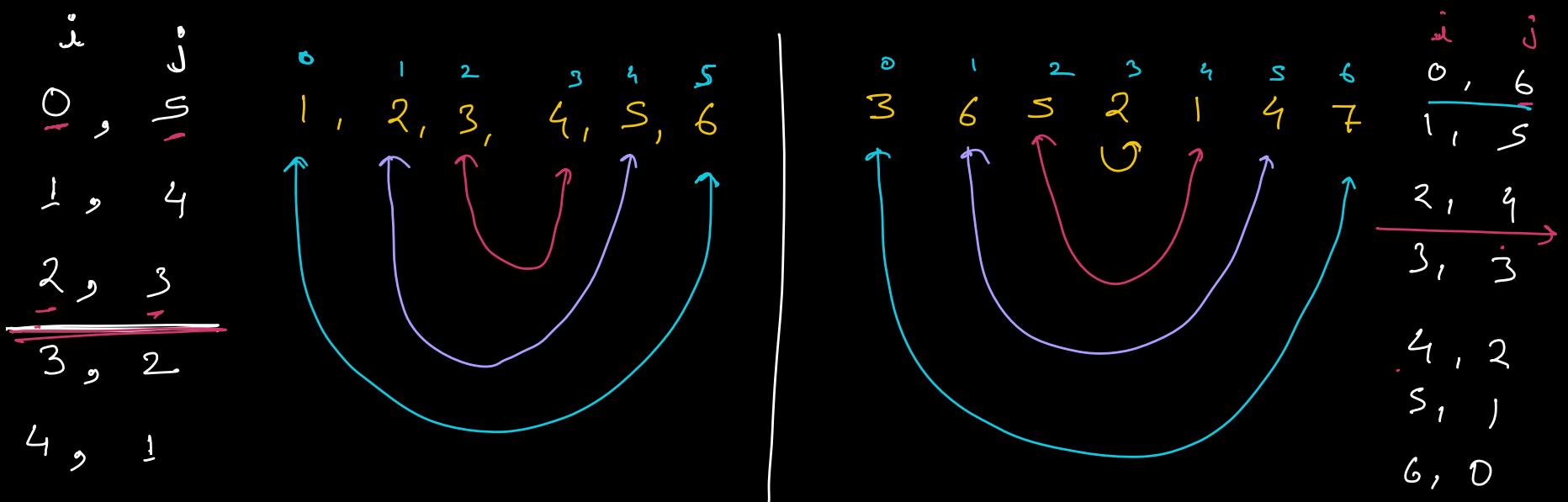
Algorithm
MS

Q Given an array of size N.

Reverse this array using constant extra spaces

A: $\begin{matrix} 0 \\ -1, 4, 7, 6, 5, -2, 12, 8, 10 \end{matrix}$

A: $\begin{matrix} 10, 8, 12, -2, 5, 6, 7, 4, -1 \end{matrix}$



```
reverse( int arr[], int N ) {
```

- $i = 0;$
- $j = N-1;$

```
    while ( i < j ) {
```

 // Swap $arr[i], arr[j]$

$i++;$
 $j--;$

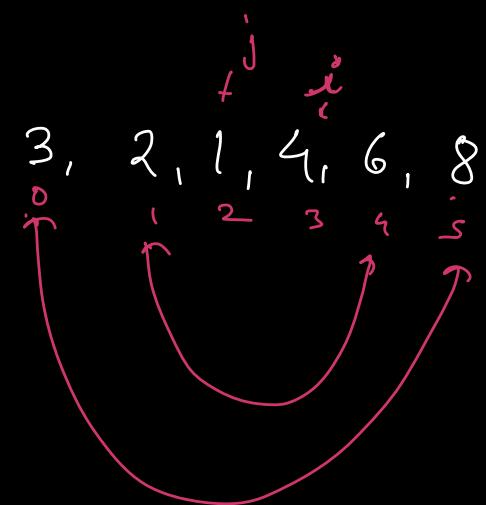
```
}
```

```
}
```

$$\# \text{No of iterations} = \frac{N}{2}$$

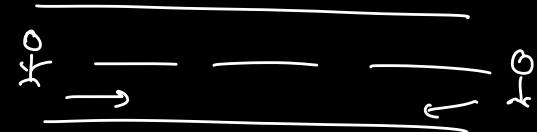
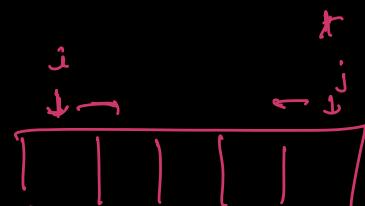
TC: $O(N)$

SC: $O(1)$
(Extra)



→ 8, 6, 4, 1, 2, 3

HW * Using a temp var]
Without using a var.]



Break till $\underline{10}$: 50 p

Q. Given an array of size N & two indices s & e.

Reverse the elements of the array from s to e. [$s \leq e$]

A: -⁰ 3, ¹ 4, ² 2, ³ 8, ⁴ 7, ⁵ 9, ⁶ 6, ⁷ 2, ⁸ 10
 $s = 3$, $e = 7$

(O(1) extra space)

A: -3, 4, 2, ² 6, 9, 7, ⁸ 10

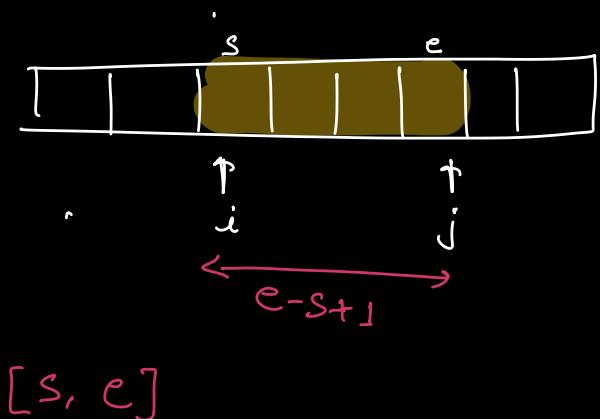
reverse (int arr [], $\underline{\text{int}}$ N, $\underline{\text{int}}$ s,
 $\underline{\text{int}}$ e)

- i = s;
- j = e;

while (i < j) {
 // Swap arr [i], arr [j]

 i++;
 j--;

}



$$\# \text{ of iterations} = \frac{(e - s + 1)}{2}$$

Worst case

$$S = 0$$

$$e = N - 1$$

TC : $O(N)$

SC : $O(1)$

Amazom

Ola

Adobe

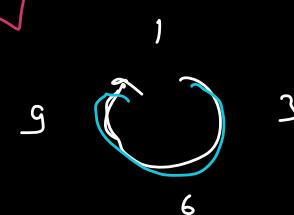
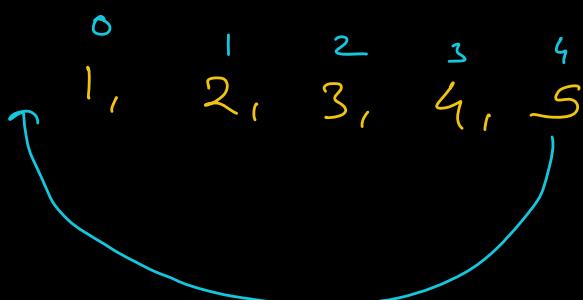
MS

Q Given an array of size N . Rotate this array

in Clockwise direction by K times.

[Using Constant extra space]

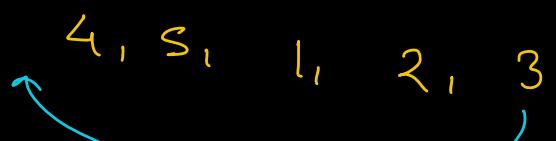
($K \leq N$)



$K=1$



$K=2$



$K=3$



K

0 1 2 3 4 5 6
3, -2, 1, 4, 6, 9, 8

L

8 3, -2, 1, 4, 6, 9

2

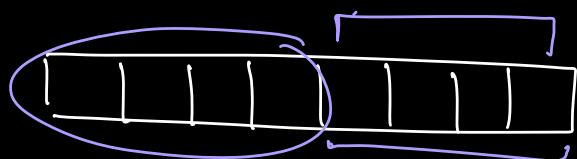
9 8 3, -2, 1, 4, 6

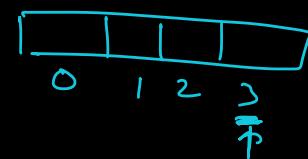
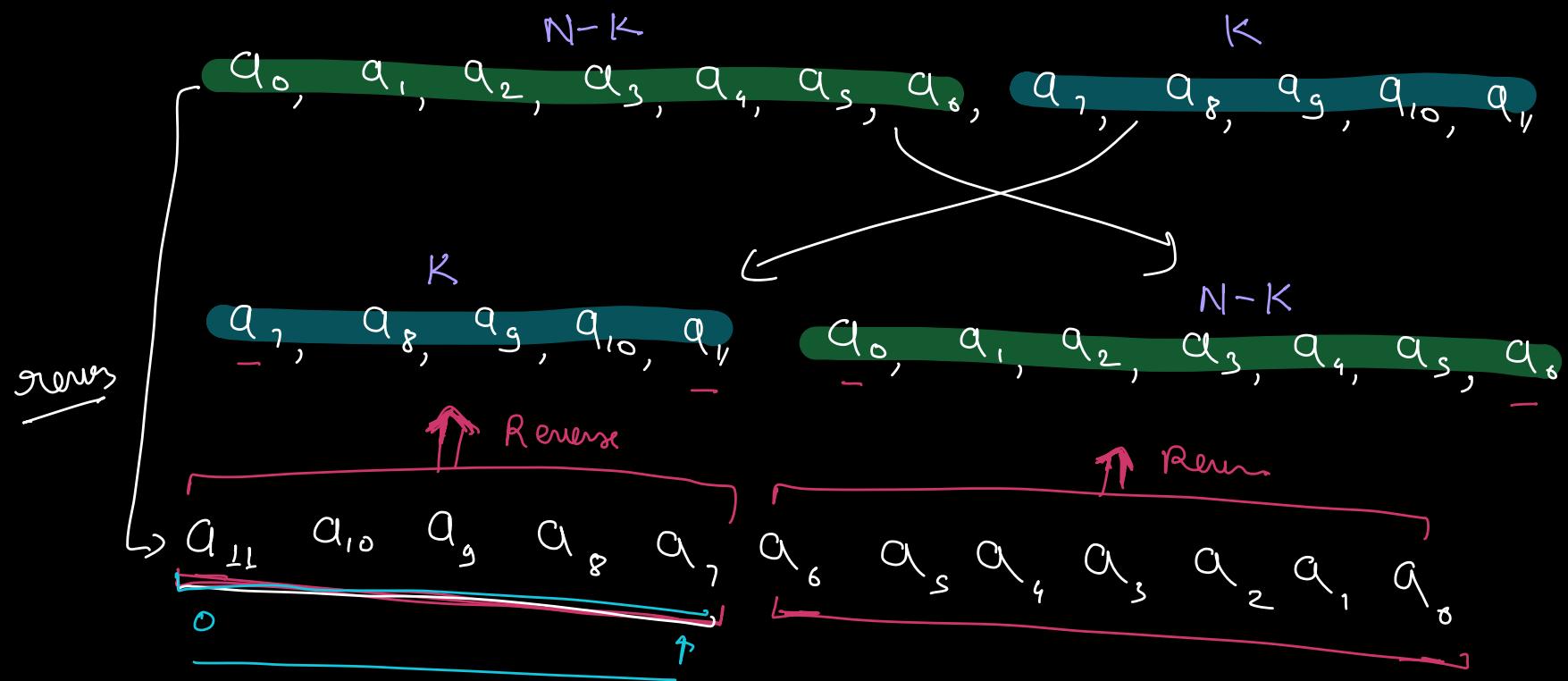
3

6 9 8 3, -2, 1, 4

4

4 6 9 8 3, -2, 1





Rotate (arr, K) {

// Reverse complete array

reverse (0, n-1);

// Reverse first K elements

reverse (0, K-1);

// Reverse the remaining N-K

reverse (K, N-1);

}

$\frac{N}{2}$

$\frac{(N-K)}{2}$

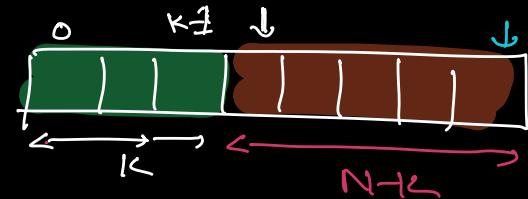
No of iteration

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

= N

TC: O(N)

SC: O(1)
(Extra)



$$N=6 \quad K=6$$

A	a_0	a_1	a_2	a_3	a_4	a_5	$6, \underline{12} \dots 60$ $7, 13$. $8, 14$ $9, 15$ $10, 16$ $11, 17$
R_0	a_0	a_1	a_2	a_3	a_4	a_5	
R_1	a_5	a_0	a_1	a_2	a_3	a_4	
R_2	a_4	a_5	a_0	a_1	a_2	a_3	
R_3	a_3	a_4	a_5	a_0	a_1	a_2	
R_4	a_2	a_3	a_4	a_5	a_0	a_1	
R_5	a_1	a_2	a_3	a_4	a_5	a_0	
R_6	a_0	a_1	a_2	a_3	a_4	a_5	

$$N = 6$$

$$K = 65$$

$$\begin{array}{r} 60 \\ \hline 5 \end{array}$$

$$\boxed{\begin{array}{l} \text{if } K > N \\ K = K \% N \end{array}}$$

Quiz

$$a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5$$

$$K = 9, N = 6$$

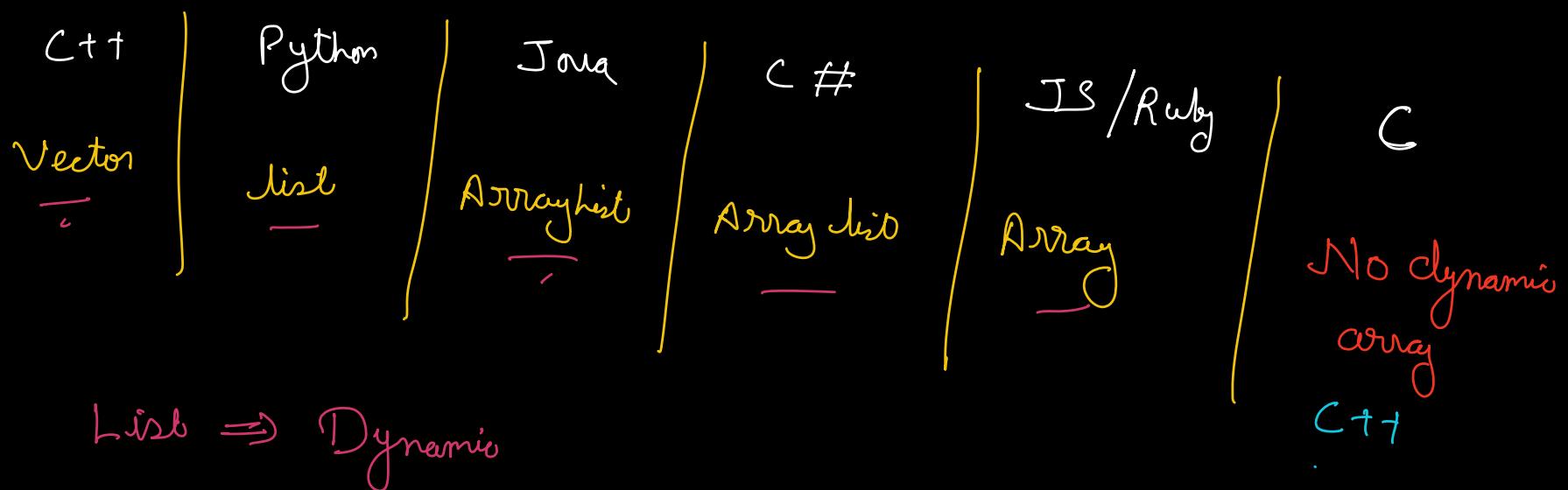
$$9 \% 6 = \underline{\underline{3}}$$

int arr[10]

→

~~Dynamic Array~~

→ Array's size is not fixed
→ dynamic



List <int> a; $\Rightarrow a.size() \rightarrow 0$ • size()

a[1] \rightarrow error

a.add(10); $\rightarrow a.size() \rightarrow 1$

a.add(21); \rightarrow

10	21
0	= 1

a.size() $\rightarrow 2$

for (i = 0; i < a.size(); i + 1) {

}

• add / insert $\Rightarrow O(1)$

• size() $\Rightarrow O(1)$