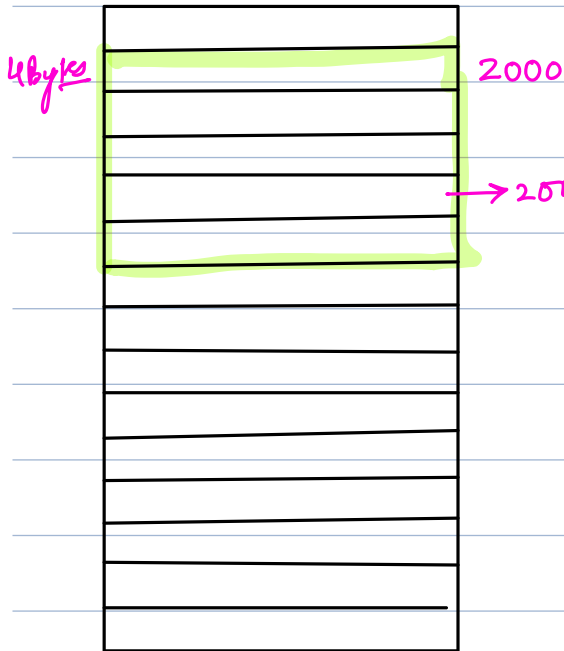
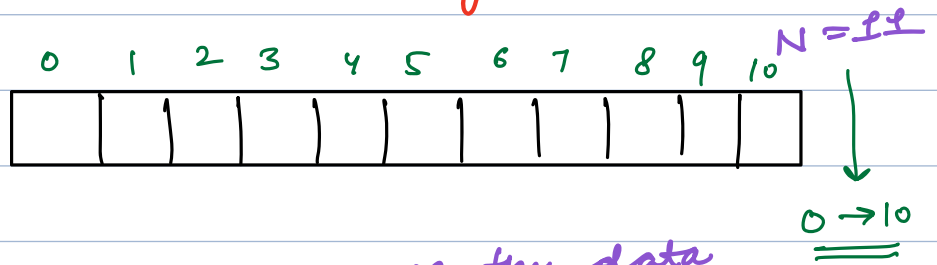


Arrays ?

list / collection of homogenous items
↓
contiguous similar



int arr[3]; size is fixed
static
Dynamic Arrays



access the data
↓
index

int arr[3];

0 → N-1

$i \rightarrow arr[i] \rightarrow O(1)$
↑

Base address + $i \times \text{size}$

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

```
    print(arr[i]);
```

```
}
```

T.C: $O(N)$

Q Given the array of size N . Find count of elements which has at least one greater element than itself.

Ex 1:

-3	-2	6	8	4	8	5
----	----	---	---	---	---	---

 ans = 5

Ex 2:

2	5	1	4	8	0	8	1	3	8
---	---	---	---	---	---	---	---	---	---

ans = 7

maximum element doesn't have any greater element.

$$\text{ans} = N - \text{freq/count of max element}$$

- 1) Find the max element
- 2) Find count of max element
- 3) ans = N - count

mx = ~~A~~ / arr[0] / INT_MIN

N {
for (i = 0; i < N; i++)
{
 mx = max(arr[i], mx);
}
}

if (arr[i] > mx)
 mx = arr[i]

int cnt = 0;

N {
for (int i = 0; i < N; i++)
{
 if (arr[i] == mx)
 cnt++;
}
}

ans = N - cnt;

2N

T.C: $O(N)$
S.C: $O(1)$

HW

↓
Try with
one loop

Q Given an array of N elements. Find count of pairs

(i, j) where i & j are indices such that

$$\text{arr}[i] + \text{arr}[j] = \textcircled{k} \xrightarrow{\substack{\text{int} \\ \text{given}}} \quad \begin{array}{l} i \neq j, i < j \\ \textcircled{2, 4} \\ 4, 2 \end{array}$$

Ex 1:-

0	1	2	<u>3</u>	4	<u>5</u>	6	
8	-2	1	4	3	6	8	$k=10$

$3, 5 \rightarrow 4+6=10$

Ex 2:-

0	1	2	3	4	5	6	7	8	9
3	5	2	1	-3	7	8	15	6	13

$k=10$

$0, 5$
 $2, 6$
 $4, 9$

$\text{ans} = 3$

0	1	2	3	4	5	6	7	8
2	7	3	14	6	1	0	10	14

$k=20$

$3, 4$
 $4, 8$

$\text{ans} = 2$

B.F:- consider all possible/valid pairs

$N = 4$

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

$i \neq j$
 $i < j$

$\text{int cnt} = 0;$

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

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

if ($\text{arr}[i] + \text{arr}[j] == k$)
 $\text{cnt}++;$

}

}

i	j	
0	$1 \rightarrow N-1$	$N-1$
1	$2 \rightarrow N-1$	$N-2$
2	$3 \rightarrow N-1$	$N-3$
3		.
4		.
...		.
$N-1$	$N \rightarrow$	<u>0</u>

$$1 \rightarrow 2 \Rightarrow \frac{n \times (n+1)}{2}$$

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

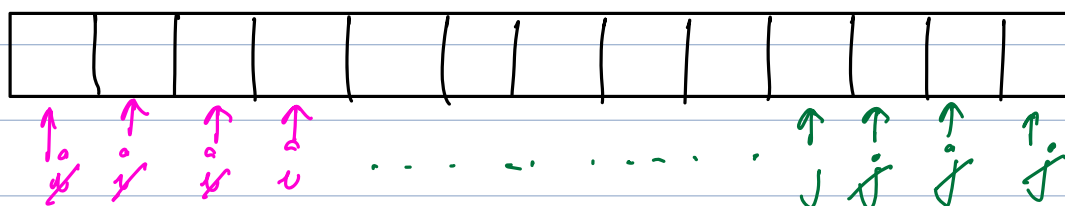
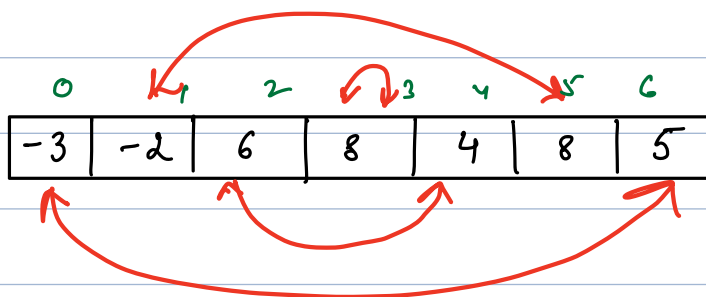
T.C: $O(N^2)$
 S.C: $O(1)$

• array \rightarrow size N . Reverse the array.

No extra space
allowed.

Ex 1:

	0	1	2	3	4	5	6
	-3	-2	6	8	4	8	5
ans =	5	8	4	8	6	-2	-3



```
void reverse( int arr[N], int N)
{
    int i = 0, j = N-1;
    while ( i <= j )
    {
        swap( arr[i], arr[j] );
        i++;
        j--;
    }
}
```

iterations $\equiv \underline{N/2}$

T.C: $O(N)$
S.C: $O(1)$

Q reverse the range $[l, r] \rightarrow$ index Based

void reverse(int arr[N], int l, int r)

{

int ~~i~~ = ~~0~~, ~~j~~ = ~~N-1~~;

while (i <= j)

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

swap(arr[i], arr[j]);

i++;

j--;

}

}

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

Ex 2:

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

10:35 pm

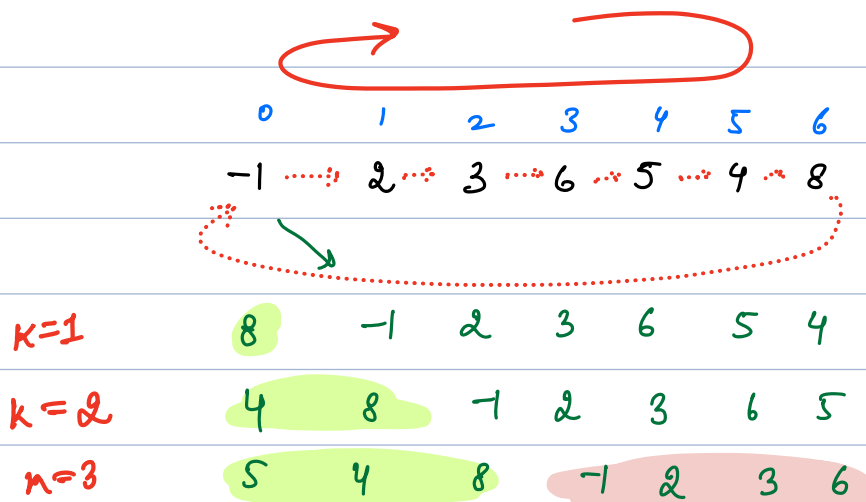
10 min break

Q given the array of size N . Rotate your array (K) times ^{give} clockwise.

$K \leq N$

No extra space allowed.

Google
Microsoft
Amazon



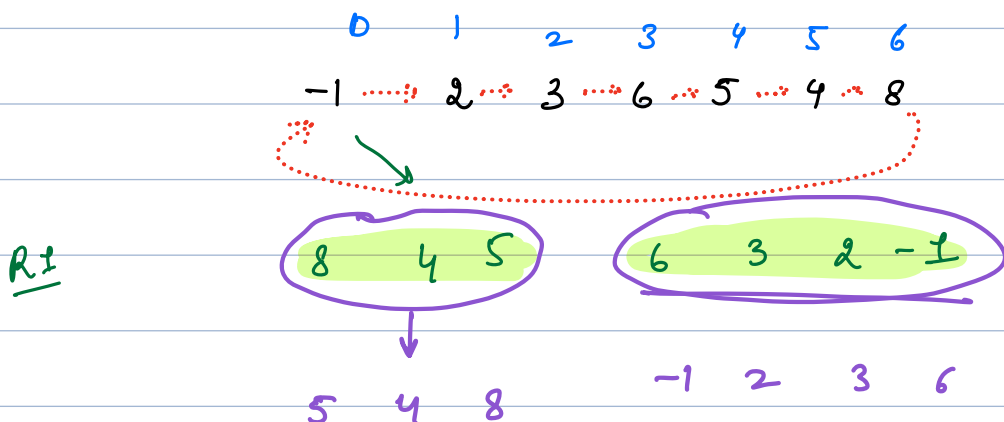
B.F

one rotation? $\rightarrow N * K$
shift every element to right

$O(N * K)$

$\approx O(N^2)$

• last k elements are coming towards start



① Reverse the complete array

reverse (0, N-1) \rightarrow N/2

② Reverse first k elements

reverse (0, k-1) \rightarrow k/2

③ Reverse remail N-k ele

reverse (k, N-1) \rightarrow N-k/2

T.C: $O(N)$

S.C: $O(1)$

N

k = N ?

k=0 -1 2 4 -2 6

k=5

k=1 6 -1 2 4 -2

k=2 -2 6 -1 2 4

k=3 4 -2 6 -1 2

k=4 2 4 -2 6 -1

k=5

N=5

k=73

73/.5 = 3

k > N

\Rightarrow k % N times

k = k % N

How dynamic arrays are implemented?



default size = 5

arr \rightarrow 10 \rightarrow 20 \rightarrow 40 \dots

static array
 \downarrow
fixed size (initially)

inserts $\rightarrow O(N)$

avg
insert $\rightarrow O(1)$

Amortized TC

A

$\text{₹ } 20 \times 300$

$\text{₹ } 6000$

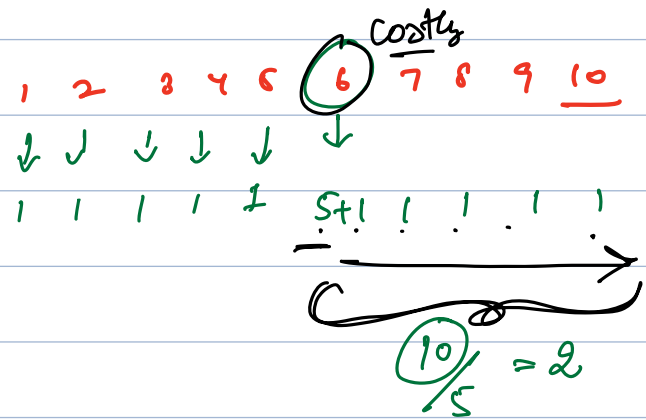
B

$\text{₹ } 6000$

300

$\frac{6000}{300} = \text{₹ } 20$

	1
	2
	3
	4
	5
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10



8 4 4 8 3 0 5 6 6 4