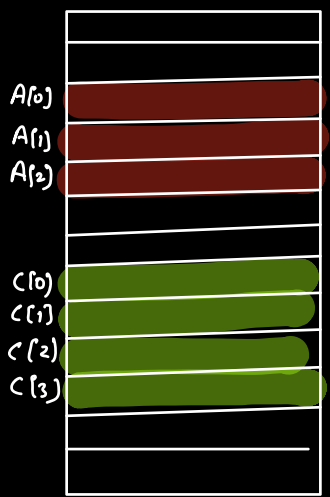
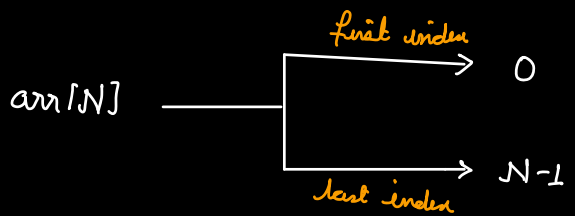
Array : List / Sq of homogeneous items

```
int [ ] A = new int [3];
int [ ] B = new int [20]; ✗
int [ ] C = new int [4];
int [ ] D = new int [3]; ✗
```





Q A : [5, -4, 8, 9, 10] (indices 0, 1, 2, 3, 4)

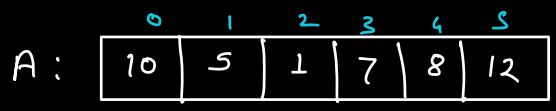
1st element ⟶ A[0]

5th element ⟶ A[4]

Print all elements

O(N)
```
for ( i = 0; i < N; i++) {
        print ( A[i] );
}
```

A[i] ⟶ O(1)



A[5] ⟶ O(1)

A[3] ⟶ O(1)

Paytm

Q Given an array of size N. Count the no. of elements which have at least one element greater than itself

-3, 2, 6, 8, 4, 8, 5 ⟶ 5

7 - 2

Quiz

2, 5, 1, 4, 8, 0, 8, 1, 3, 8 ⟶ 7

10 - 3

Observations

Only the largest element will not be having a greater element

ans ⟶ N − Count of the max element

Step I : Find the max no. → N iterations

Step II : Iterate over array & count the frequency of max no. (C_{max}) → N iterations

Step III : ret $N - C_{max}$ → 1 "Nothing is free"

iterations ⇒ $N + N + 1$

$= 2N + 1$

TC : $O(N)$

SC (Extra space) : $O(1)$ $O(N)$

HW: do Step I & II in a single loop.

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

Google Facebook Amazon MS

Return true if there exists a pair A[i], A[j] such that A[i] + A[j] = K (i != j)

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

K : 10 ⟶ True

K : 6 ⟶ True

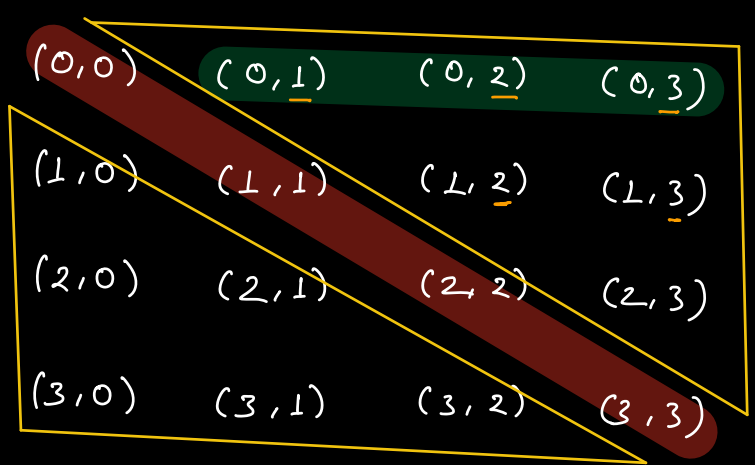
K : 16 ⟶ False

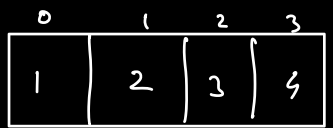
A : [2, 7, 3, 14, 6, 1, 0, 10, 14] ⟶ 2

Brute Force

Check all pairs & compare sum with K.

N = 4





(0, 2) A[0] + A[2]

(2, 0) A[2] + A[0]

```
for(i=0; i<N; i++){
    for(j=0; j<N; j++){
        if(i != j){
            if(A[i] + A[j] == k){
                ret tru;
            }
        }
    }
    ret false;
```

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) {
            return true;
        }
    }
}
ret false;
```

i	j [i+1, N-1]	iterations
0	[1, N-1]	N-1
1	[2, N-1]	N-2
2	[3, N-1]	N-3
3	[4, N-1]	N-4
⋮	⋮	⋮
i	[i+1, N-1]	N-i
⋮	⋮	⋮
N-2	[N-1, N-1]	1

$1 + 2 + 3 + \cdots + (N-3) + (N-2) + (N-1)$

$a = 1$, $d = 1$, no of terms $= (N-1)$

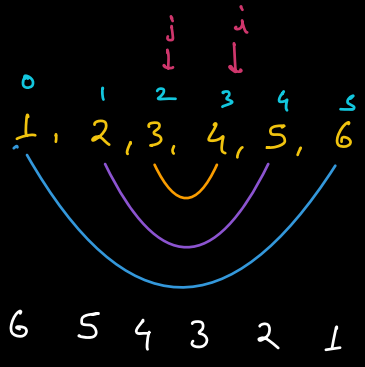
iterations $\Rightarrow \frac{N(N-1)}{2}$

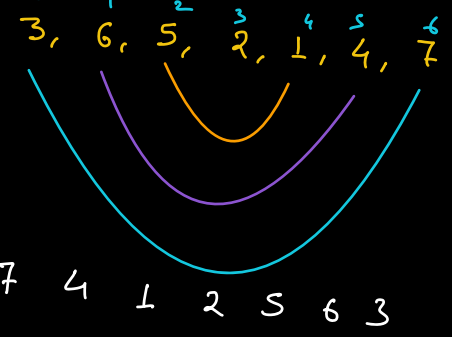
TC : $O(N^2)$

SC (Extra) : $O(1)$

Q Given an array. Reverse it without using any extra space (constant amount of space, SC: O(1)).

Ayo





i	j
0	5
1	4
2	3
---	---
3	2
4	1
5	0

i < j

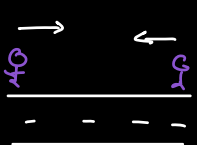
X

i	j
0	6
1	5
2	4
---	---
3	3
4	2
5	1
6	0

i < j

X

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



iterations = N/2

TC : O(N)

SC : O(1)

Q PayTM Screening

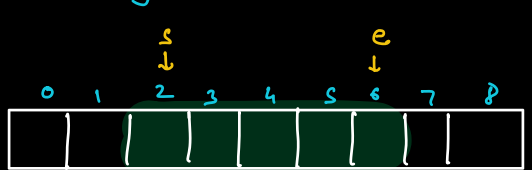
Given an array of size N & two indices s & e. (s<=e)

Reverse the elements of the array from index s to e.

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

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

```
void reverse (A[], N, s, e) {
    i = s;
    j = e;
    while (i < j) {
        swap (A[i], A[j]);
        i++;
        j--;
    }
}
```



[s, e] ⇒ e - s + 1

iterations → $\frac{(e - s + 1)}{2}$

Worst case

e → N-1

s → 0

iterations = N/2

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

Amazon
Ola
Adobe
MS

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

Rotate the array in clockwise direction [right to left] K times. ($K < N$)

Without using any extra spaces

SC : O(1)

A : 1, 2, 3, 4, 5 (0, 1, 2, 3, 4)

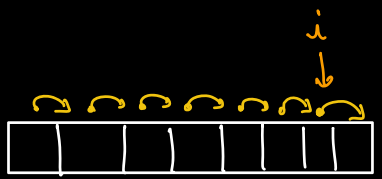
K = 1 : 5, 1, 2, 3, 4

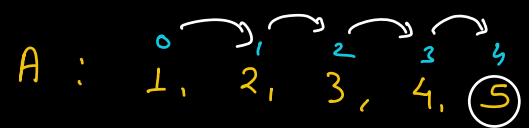
K = 2 : 4, 5, 1, 2, 3

K = 3 : 3, 4, 5, 1, 2

A[i] → A[i+1]

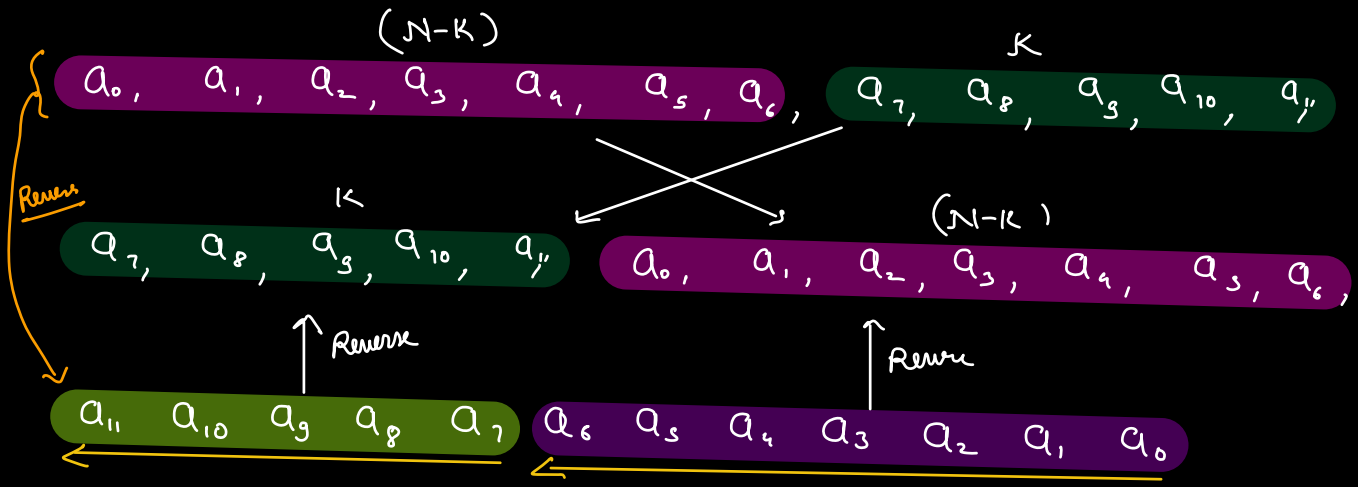
i--





TC : O(NK) ⟶ O(N^2)

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



Step I Reverse complete array

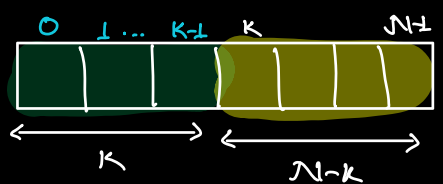
Step II Reverse the first K elements

Step III Reverse the last (N-K) elements

```
void rotate (A[], N, K) {

    reverse (A, 0, N-1);   ⟶ N/2
    reverse (A, 0, K-1);   ⟶ K/2
    reverse (A, K, N-1);   ⟶ (N-K)/2

}
```



iterations $= \frac{N}{2} + \frac{K}{2} + \frac{N-K}{2} \Rightarrow$

$\rightarrow \frac{N}{2} + \frac{K}{2} + \frac{N}{2} - \frac{K}{2} \Rightarrow N$

TC : O(N)

SC : O(1)

$K > N \longrightarrow K = K \% N$

$\longrightarrow N-1$

	0	1	2	3	4
A :	1,	2,	3,	4,	5

K = 1 :	5, 1, 2, 3, 4	K = 6, 11, 16
K = 2 :	4, 5, 1, 2, 3	K = 7, 12, 17
K = 3 :	3, 4, 5, 1, 2	K = 8, 13, 18
K = 4 :	2 3, 4, 5, 1	K = 9, 14, 19
K = 5 :	1 2 3, 4, 5	K = 10, 15, 20

K ka	1	2	3	4	5	6	7	8	9	10	11	12	13	14	10
K % N	1,	2,	3,	4,	0,	1,	2,	3,	4,	0,	1,	2,	3,	4,	0

K % N

$6 \longrightarrow 5 + 1$

$7 \longrightarrow 5 + 2$

30 mins

```
for (i = N; i > 0; i = i/2) {
    for (j = 0; j < i; j++) {
        c = c+1;
    }
}
```

$$N + \frac{N}{2} + \frac{N}{4} + \cdots + 1$$

GP

No of terms ? $\rightarrow \log_2 N + 1$

$32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \Rightarrow$ $N \longrightarrow 1$