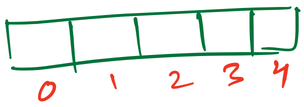
Introduction to Arrays

Array ?

↳ list of homogeneous items

contigous ↳ same type

↳ Size is fixed → static size

N = 5 

index to access a value in a array

Indexing in the array of size N starts from 0 to N-1.

```
int a[5];
```

a[5]

```
for (i=1; i<=5; ++i) {
    a[i] = x;
}
```

```
int a[10];
```

a[0], a[1], ... , a[9], ~~a[10]~~, ~~a[11]~~

Time complexity of accessing any value of an array is $\boxed{O(1)}$

a[n]   a[i] = ?   constant ⇒ O(1)

To access all the elements it will take O(N)

Traversal in Array

```
int a[n];

for (i=0; i<n; i++) {
        a[i]
}
```

iterations = N

T.C = O(N)

Question 1

Given an array of size N. Find count of elements which has atleast one greater element than itself.

eg   1 4 5 3 -1 5 4

ans = 5

eg   2 5 1 4 8 0 8 1 3 8

1. find the max element
2. Count the occurances of max element
3. ans = N - count

```
int mx = ~~0~~;     we can have -ive values
    mx = a[0];
```

```
      for ( i=1 ; i<n; ++i ) {
N-1       mx = max(mx, a[i]);
      }
      int count=0;
      for (i=0; i<n; ++i) {
N         if (mx == a[i])
              ++count;
      }
      ans = n - count;
```

```
if(a[i]>mx)
    mx=a[i];
```

T.C = O(N)

S.C = O(1)

H.W: Do it in one FOR loop.

eg a[3]={-3 -1 -2}

mx=0 => mx=0

Question 2

Given an array of N elements. Find count of pair (i,j) where i,j are indicies such that

a[i]+a[j] = K (given)   i<j

eg a[6]={1 5 6 2 4 3}   K=7

1	5	6	2	4	3
0	1	2	3	4	5

(0,2) (1,3) (4,5) $\boxed{ans = 3}$

$a[3] = \{ \underset{0}{1} \quad \underset{1}{6} \quad \underset{2}{6} \}$ $K = 7$

$\boxed{ans = 2}$

(0,1) (0,2)

$a[\,] = \{ \underset{0}{3} \quad \underset{1}{5} \quad \underset{2}{2} \quad \underset{3}{1} \quad \underset{4}{-3} \quad \underset{5}{7} \quad \underset{6}{8} \quad \underset{7}{15} \quad \underset{8}{6} \quad \underset{9}{13} \}$ $\underline{K=10}$

(0,5) (2,6) (4,9) $\boxed{ans = 3}$

a[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)

discard

```
count=0
for (i=0;  i<n;  ++i) {
    for (j=i+1 ; j<n;  ++j ) {
        if ( a[i] + a[j]==K )
            ++count;
    }
}
```

i	j=[i+1,n-1]	iteration
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	NA	0

$T.C. = O(N^2)$

$S.C = O(1)$

$1+2+3+.....+n-1$

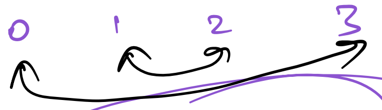
$$\frac{n \times (n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Question 3

Given an array of size N. Reverse the array without using extra space.

eg

$\{1 \quad 2 \quad 3 \quad 4\} \xrightarrow{reverse} \{4 \quad 3 \quad 2 \quad 1\}$



0th → 3rd

1st → 2nd

2nd → 1st

3rd → 0th

0th ⟷ 3rd

$a[0] \longleftrightarrow a[n-1]$ $\qquad 0+n-1 = n-1$

$a[1] \longleftrightarrow a[n-2]$ $\qquad 1+n-2 = n-1$

$\vdots$

$a[i] \longleftrightarrow a[\underset{n-i-1}{\underset{\Downarrow}{j}}]$ $\qquad i+j = n-1$

$j = n-i-1$

```
for(i=0; i<n; ++i){
    swap(a[i], a[n-i-1]);
}
```

NOT WORK

	1	2	3	4
	0	1	2	3
i=0	4	2	3	1
i=1	4	3	2	1
i=2	4	2	3	1
i=3	1	2	3	4

```
i=0, j=n-1;
while(i<j){
    swap(a[i], a[j]);
    ++i
    --j
}
```

WORKS

iterations = $n/2$

T.C: O(N)

S.C: O(1)

What if we have to reverse only some part of array.

[2, 6]

1	2	3	4	5	6	7	8	9	10

1 2 7 6 5 4 3 8 9 10

[start, end]

```
i = start , j = end;
while (i < j ) {
    swap (a[i] , a[j]);
    ++i;
    --j;
}
```

Break : 10:33 - 10:43 PM

Question 4

Given an array of size N. Rotate your array K times clockwise. K<N.

eg

1 2 3 4 5 $\boxed{K=3}$

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

extra space →

3	4	5	1	2
0	1	2	3	4

(i+K)%n

T.C = ~~O(1)~~

~~O(K)~~

O(N)✓

S.C = O(N)✓

~~O(1)~~

```
int temp[n];
for(i=0; i<n; ++i) {
    j= (i+K)%n;
    temp[j] = a[i];
}
for (i=0; i<n; ++i) {
        a[i] = temp[i];
}
```

i=0, j= 0+3=3%5=3

i=1, j= 4%5=4

i=2, j= 5%5=0

i=3, j= 6%5=1

i=4, j=7%5=2

1   2   3   4   5   K=2

⇓

4   5   1   2   3

reverse

5 4   3 2 1

reverse

C reverse   U reverse

4 5 1 2 3

To shift element K times clockwise ⇒

iterations

reverse (0 , N-1) ⇒ $N-1/2$

reverse (0 , K-1) ⇒ $K-1/2$

reverse (K , N-1) ⇒ $N-1-K/2$

$$\frac{N-1+\cancel{K}-1+N-1-\cancel{K}}{2}$$

SC : O(1)

$$\frac{2N-3}{2}$$

$= N - \frac{3}{2}$ ⇒ O(N) TC

```
void reverse( int a[], int start, int end){
      i= start , j= end;
      while (i< j ) {
            swap (a[i] ,a[j]);
            ++i;
            --j;
      }
}
```

```
reverse( a , 0 , n-1 );
```

```
reverse ( a , 0 , K-1 );
revers ( a , K , n-1 );
```

Dynamic Array : Array with variable size

C++	Java	Python	JS/Ruby
vector	ArrayList	list	array

append()/ add()/ insert()/ push-back()

Doubt

```
for (i=3; i<n/3; i+=3)
    for (j=2; j<n/2; j+=2)
```

$i = 3, 6, 9, \ldots n/3 \Rightarrow n/9$ $\qquad n/9 \times n/4$

$j = 2, 4, 6, \ldots n/2 \Rightarrow n/4$ $\qquad \boxed{\frac{n^2}{36}}$

$O(n^2)$