Arrays

Java

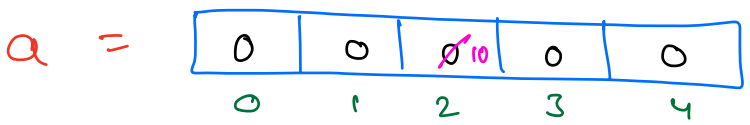
Syntax

```
int [] a = new int[5]
```

Python

Syntax

```
a = [0] * 5
```



a[2] = 10

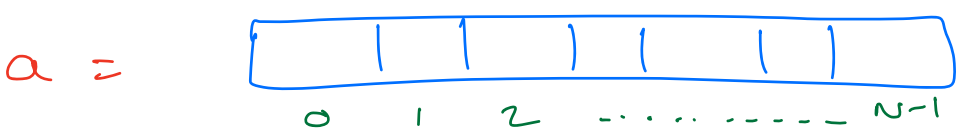
How to declare an array of size N ?

Java

```
int [] a = new int[N];
```

Python

```
a = [0] * N
```



Passing array to functions

```
void printArr (int [] a) {
    int n = a.length;
    for(int i=0; i<n; i++) {
        sop (a[i]);
    }
}
```

```
def printArr (a):
    n = len(a)
    for i in range (n):
        print (a[i])
```

Quiz 1

First = 0

Last = N-1

Quiz 2

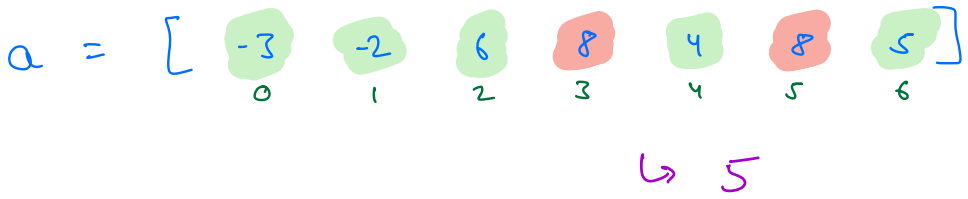
arr[5] = 5, -4, 8, 9, 10
(0, 1, 2, 3, 4)

Sum of 1st and 5th element

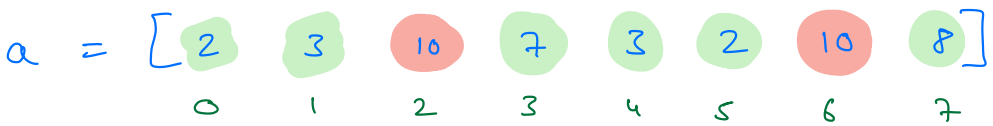
arr[0] + arr[4]

Q1.

Given N array elements , count no. of elements having atleast 1 element greater than itself.



↳ 5



Quiz 3

↳ 6

↳ 7

Observations

→ For every max element, there won't be any element greater than it.

→ For every other element, there will be atleast one element greater than it.

1) Iterate & get the max value
2) Iterate & get the count of elements != max

Pseudocode

```
int countGreater(int arr[]) {

    // find the max
    maxVal = -∞                    → Integer.MIN_VALUE
                                   → float ("-inf")
    for(i=0; i<n; i++) {
        if ( arr[i] > maxVal)
            maxVal = arr[i]
    }                                          } N

    c = 0
    for( i=0; i<n; i++) {
        if ( arr[i] != maxVal)                 } N
            c++
    }
    return c

}
```

Total iterations
= 2N

Any extra space ?
↓
No

Time - O(N)
Space - O(1)

Q2 Given N array elements, check if there exists a pair (i, j) such that

bool

$$arr[i] + arr[j] == k \quad \&\& \quad i != j$$

Note : i & j are index values , k is given sum

a = [3 -2 1 4 3 6 8]
0 1 2 3 4 5 6

k = 10

i=3 , j=5 4 + 6 = 10 ⇒ True

a = [2 4 -3 7]
0 1 2 3

k = 5

No pair ⇒ false

---

a = [3 5 2 7 3]
0 1 2 3 4

k = 6

Quiz 4

3 + 3 = 6

i: 0 j:4 ⇒ True

---

a = [2, 5, 9, 6]
0 1 2 3

k = 18

⇒ false

~~2~~ ~~7~~ 9
2 2

Idea 1

→ Run a loop with i for all indices

→ Run a nested loop with j over all indices

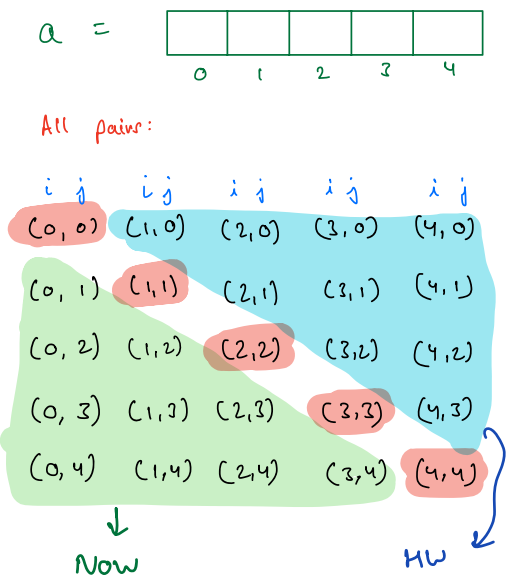
Time - $O(N^2)$

Space - $O(1)$

```
boolean checkSum(int arr[], int k) {
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            if( arr[i] + arr[j] == k  and  i!=j)
                return true
    }   }

    return false
}
```

arr[i] + arr[j] = arr[j] + arr[i]



Idea 2

```
boolean checkSum(int arr[], int k) {
    for(i=0; i<N-1; i++) {
        for( j=i+1; j<N; j++) {
            if( arr[i] + arr[j] == k )
                return true
        }
    }
    return false
}
```

i	j	Iterations
0	1 → N-1	N-1
1	2 → N-1	N-2
2	3 → N-1	N-3
⋮		⋮
N-2		0

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

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

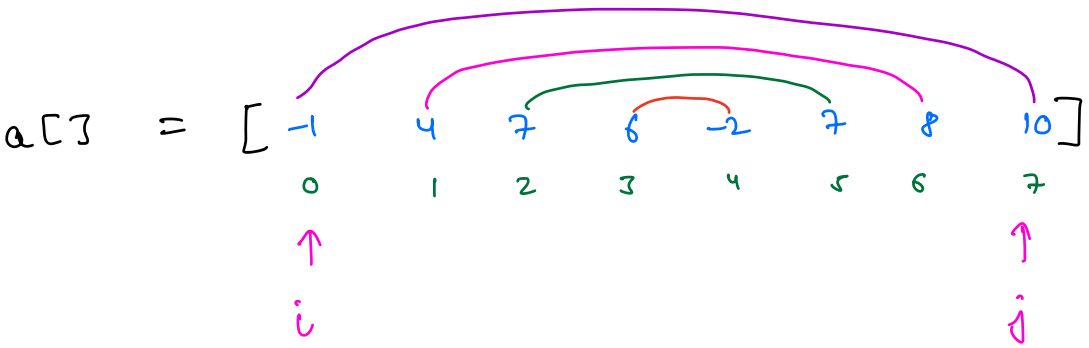
Time - $O(N^2)$

Space - $O(1)$

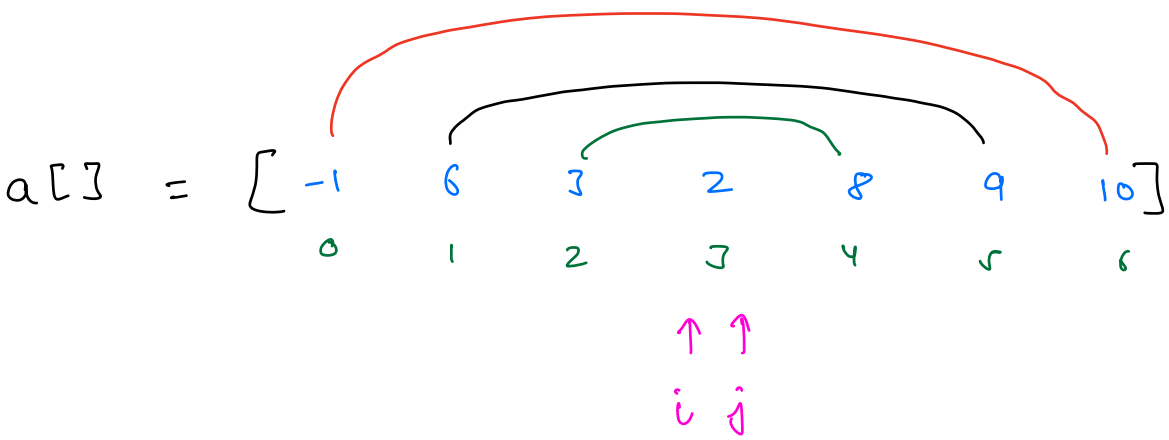
Q3 Given an array , reverse the entire array

Note : Array itself should change

Expected SC: O(1)



$i < j$



swap (A[i], A[j])

i → i+1

j → j-1

Pseudocode

```
reverseArray(int arr[]) {
        N = arr.length
        i = 0 , j = N-1
        while ( i < j ) {
            swap (arr[i], arr[j])
            i = i+1
            j = j-1
        }
}
```

Quiz 5

Time - O(N)

Space - O(1)

Iterations = N/2

Break till 10:15 PM

Q4 Given an array , and [S & E],
reverse the array from [S E], where S and E are indices.
Note : S <= E

a[] = [-3 4 2 8 7 9 6 2 10]
0 1 2 3 4 5 6 7 8

S E
[3 7]

a[] = [-3 4 2 2 6 9 7 8 10]
0 1 2 3 4 5 6 7 8

```
reversePart(int arr[], int s, int e) {

    i = s
    j = e

    while ( i < j ) {
        swap (arr[i], arr[j])
        i = i+1
        j = j-1
    }

}
```

Worst case
s = 0, e = n-1
Entire array

Quiz 6

Time - O(N)
Space - O(1)

Q5. Given an array of size N, rotate the array from last to first by k times.

Expected SC: O(1)

$k > 0$

Example

arr [7] = [3 -2 1 4 6 9 8]

0 1 2 3 4 5 6

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

Example

arr [9] = [4 1 6 9 2 14 7 8 3]

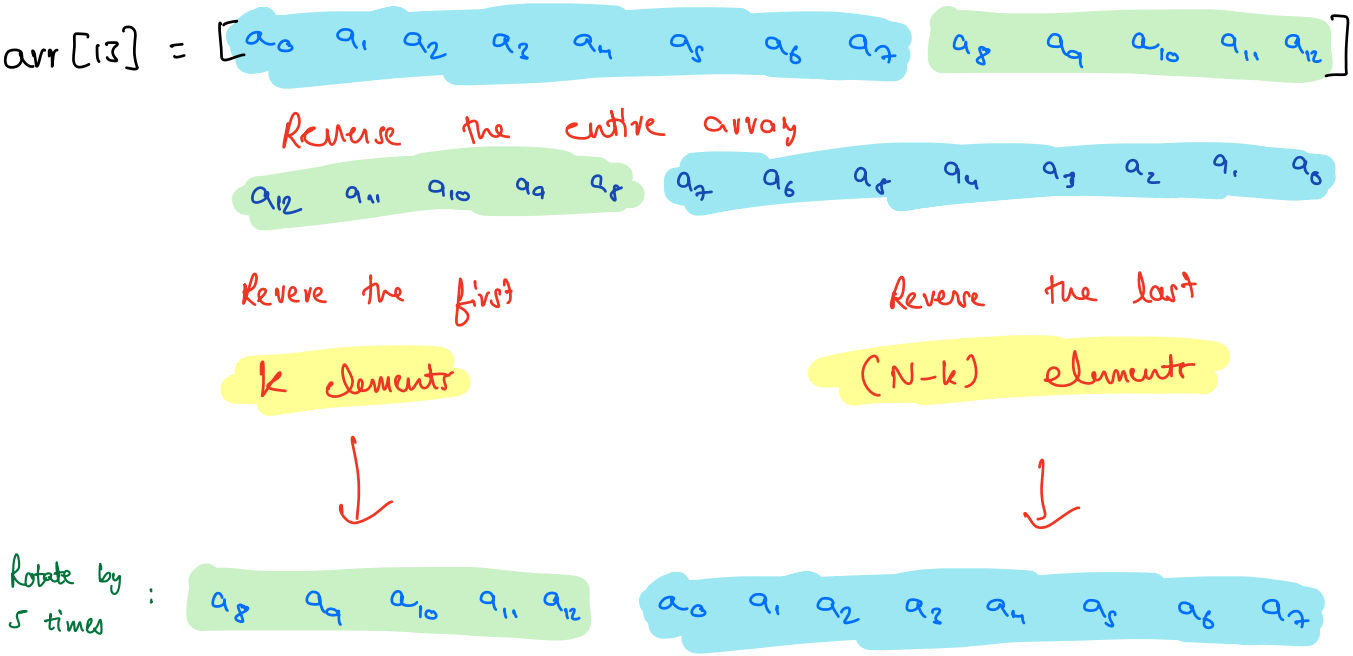
0 1 2 3 4 5 6 7 8

k = 4

arr [9] = [14 7 8 3 4 1 6 9 2]

0 1 2 3 4 5 6 7 8

Brute force - 2 loops - O(NK)



1) Reverse entire array , $s = 0$, $e = N-1$

2) Reverse first k elements , $s = 0$, $e = k-1$

3) Reverse last N-k elements , $s = k$
$e = N-1$

```
rotateTimes(int arr[], int k) {
```

n = arr.length

reverse Part (arr, 0, n-1) → N

reverse Part (arr, 0, k-1) → k

reverse Part (arr, k, N-1) → N-k

```
}
```

Total iterations = N + ~~k~~ + ~~N-k~~ = 2N

Time - O(N)

Space - O(1)

Quiz 7

What if k > N ?

N = 5

a = [5 2 9 10 3]

k = 5 → 5 2 9 10 3

k = 6 → 3 5 2 9 10

ar[6] = a_0 a_1 a_2 a_3 a_4 a_5

n = 6

		Rotate k times		
k = 0	a_0 a_1 a_2 a_3 a_4 a_5	6	12	18
k = 1	a_5 a_0 a_1 a_2 a_3 a_4	7	13	19
k = 2	a_4 a_5 a_0 a_1 a_2 a_3	8	14	20
k = 3	a_3 a_4 a_5 a_0 a_1 a_2	9	15	⋮
k = 4	a_2 a_3 a_4 a_5 a_0 a_1	10	16	
k = 5	a_1 a_2 a_3 a_4 a_5 a_0	11	17	

$k = k \% N$

```
rotateTimes(int arr[], int k) {
```

n = arr.length

k = k % N

reverse Part (arr, 0, n-1)

reverse Part (arr, 0, k-1)

reverse Part (arr, k, N-1)

```
}
```

TC & SC remain same

Doubts

Thank You

```
for (i=1; i<=n; i++) {
    for (j=1; j<=3^i; j++) {
        ——
    3
}
```

i	j	Iterations
1	$1 \to 3$	3
2	$1 \to 3^2$	3^2
3	$1 \to 3^3$	3^3
⋮		⋮
N	$1 \to 3^N$	3^N

Total =

$$3 + 3^2 + 3^3 + \dots + 3^N$$

GP

```
for ( i=0; i < 2^N ; i++ ) {
    j = i
    while (j > 0)
        j--;
}
```

i	j	Iterations
0	0 → 1	0
1	1 → 1	1 +
2	2 → 1	2 +
!	3 → 1	3 +
		. +
2^N	2^N → 1	2^N +

Total = $1 + 2 + 3 + \dots 2^N$

→ $\frac{2^N(2^N+1)}{2}$

Sum of first N = $\frac{N(N+1)}{2}$

$4^N \leftarrow \frac{(2^N)^2}{2} + \frac{2^N}{2}$

$O(4^N)$ $\Leftarrow$ $\frac{4^N}{2} + \frac{2^N}{2}$

$(2^N)^2 = 4^N$

$\Downarrow$

$(2^2)^N \Rightarrow 4^N$

$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \frac{N}{16} + \cdots$ $\quad 0$

$\frac{a(1-r^N)}{1-r}$

$r = \frac{1}{2}$,

$\frac{N\left(1-\frac{1}{2}^N\right)}{1-\frac{1}{2}} = 2N\left(1-\frac{1}{2^N}\right)$

$= 2N\left(\frac{2^N-1}{2^N}\right)$

$x = a / b$, $b == 0$ is a corner case

---

i	j	Iterations
1		5
2		5
3		5
⋮		
N		5

Total iterations

=

$5 + 5 + 5 + 5 + \ldots + 5$ (N times)

$= 5 \times N$

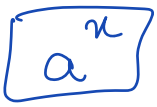
i	j	Iterations
1		1
2		2
3		3
⋮		⋮
N		N

Total iterations

$= 1 + 2 + 3 + 4 \ldots N$

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

→ log



→ Exponential

Good Night

Thank You

Monday