

# ⇒ left diagonal traversal

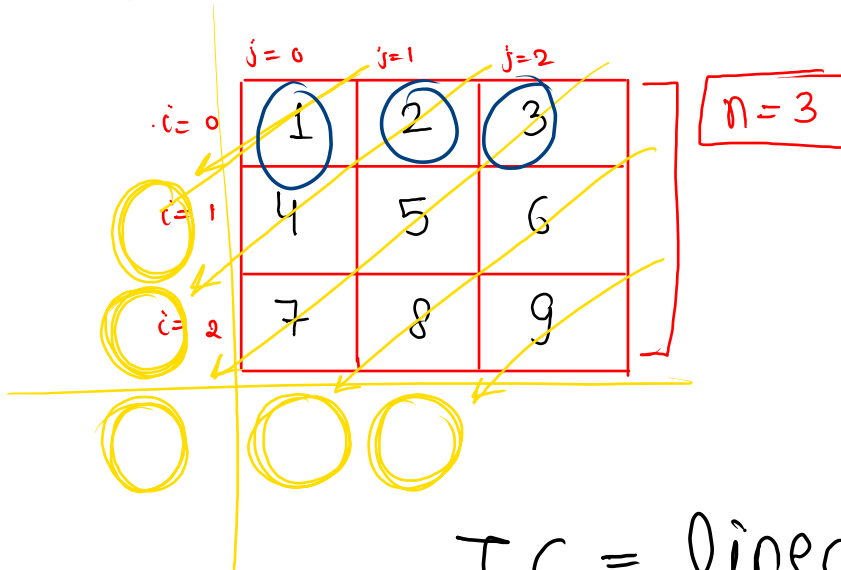
code

```
public static void leftDiagonal(int[][] arr, int n) {
```

```
    for (int gap = 0; gap < n; gap++) {
        for (int i = 0, j = gap; j >= 0; i++, j--) {
            System.out.print(arr[i][j] + " ");
        }
    }
```

```
    for (int gap = 1; gap < n; gap++) {
        for (int i = gap, j = n - 1; i < n; i++, j--) {
            System.out.print(arr[i][j] + " ");
        }
    }
```

```
}
```



ans:-

1 2 3 5 7 8 9

gap=0,  $(i,j)$   
 $(0,0)$   
 $(1,-1)$

gap=1  $(0,1)$   
 $(1,0)$   
 $(2,-1)$

gap=2  $(0,2)$   
 $(1,1)$   
 $(2,0)$   
 $(3,-1)$

gap=3  $(3 < 3) \times$

gap=1  $(i,j)$   
 $(1,2)$   
 $(2,1)$   
 $(3,0) \times$

gap=2  $(2,2)$   
 $(3,1) \times$

gap=3  $(3 < 3) \times$

T.C = linear =  $O(n \times n)$

# Transpose of Matrix of N\*N

(transpose :- when all the rows become cols and cols become rows.)

Arr =

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

→  
transpose

	0	1	2	3
0	1	5	9	13
1	2	6	10	14
2	3	7	11	15
3	4	8	12	16

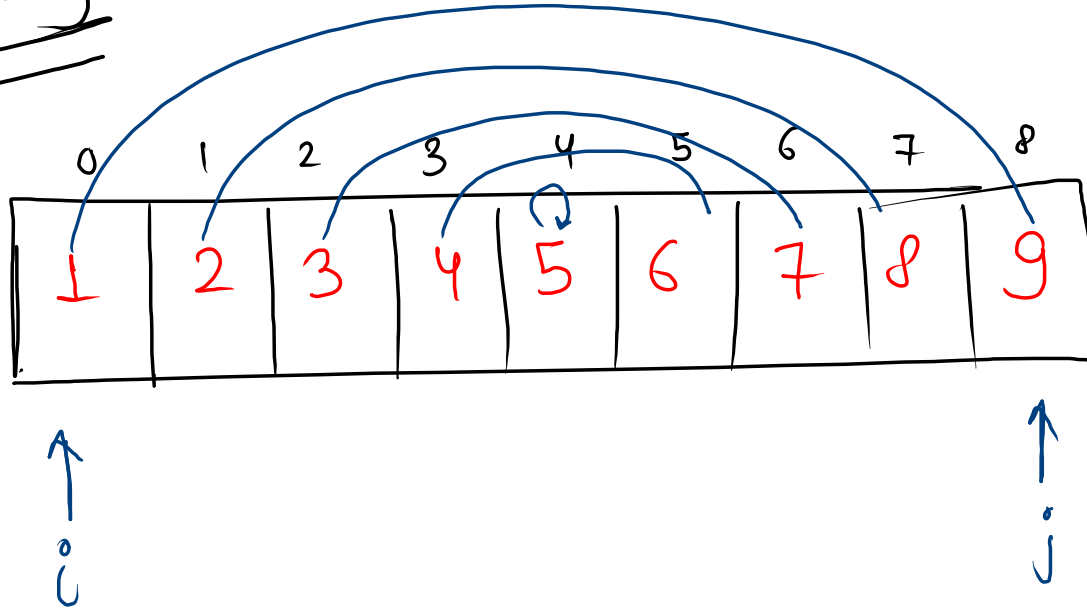
rows = i  
cols = j

$$\underline{\underline{(i, j)}} \Leftrightarrow \underline{\underline{(j, i)}}$$

Swap

Ex with  
1D array

arr



till middle

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    transpose(arr, n);
}

public static void transpose(int[][] arr, int n) {

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i > j) {
                int temp = arr[i][j];
                arr[i][j] = arr[j][i];
                arr[j][i] = temp;
            }
        }
    }

    // print
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}
```

$$T.C = O\left(\frac{n \times n}{2}\right)$$

$$\approx O(n^2)$$

linear

indexes

		→			
		0	1	2	3
0		X	X	X	X
1		O	X	X	X
2		O	O	X	X
3		O	O	O	X

$(0,0)$  X

$(0,1)$  X

$(0,2)$  X

$(0,3)$  X

$(1,0)$  ✓

$(1,1)$  X

$(1,2)$  X

$(1,3)$  X

$(2,0)$  ✓

$(2,1)$  ✓

$(2,2)$  X

$(2,3)$  X

$(3,0)$  ✓

$(3,1)$  ✓

$(3,2)$  ✓

$(3,3)$  X

$i=1, j=0$  ✓

$i=0, j=1$  ✓

# Rotate The Matrix by 90 Degree

arr

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

90°

transpose

	0	1	2	3
0	1	5	9	13
1	2	6	10	14
2	3	7	11	15
3	4	8	12	16

	0	1	2	3
3	13	14	15	16
2	9	10	11	12
1	5	6	7	8
0	1	2	3	4

reverse each row

→ reverse each row ( $n \times n$ )

arr

	13	9	5	1
$i=0$	<del>1</del>	<del>5</del>	<del>8</del>	<del>13</del>
	↑			↑
	$s_i$			$e_i$
$i=1$	2	6	10	14
$i=2$	3	7	11	15
$i=3$	4	8	12	16

```
for (int i=0; i<n; i++) {  
    1) loop until  $s_i < e_i$   
    1.1) swap( $s_i, e_i$ )  
         $s_i++$ ;  
         $e_i--$ ;  
}
```

```

① public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    rotate90(arr, n);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

```

```

② public static void rotate90(int[][] arr, int n) {
    //step1 - transpose
    transpose(arr, n);
    //step2 - reverse each row
    reverseEachRow(arr, n);
}

```

```

③ public static void reverseEachRow(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        int si = 0;
        int ei = n - 1;
        while ( si < ei ) {
            swap( arr[i], si, ei );
            si++;
            ei--;
        }
    }
}

```

```

④ public static void swap(int[] arr1d, int i, int j) {
    int temp = arr1d[i];
    arr1d[i] = arr1d[j];
    arr1d[j] = temp;
}

⑤ public static void transpose(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if ( i > j ) {
                int temp = arr[i][j];
                arr[i][j] = arr[j][i];
                arr[j][i] = temp;
            }
        }
    }
}

```

$$T.C = n^2 + n^2$$

$$= 2n^2$$

$$= \underline{\underline{O(n^2)}}$$



# Rotate The Matrix by 180 Degree

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    int[][] arr = new int[n][n];  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            arr[i][j] = scn.nextInt();  
        }  
    }  
}
```

```
    rotate90(arr, n);  
    rotate90(arr, n);
```

—————> 2 times

```
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            System.out.print(arr[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```

$$\underline{\underline{T.C = O(n^2)}}$$