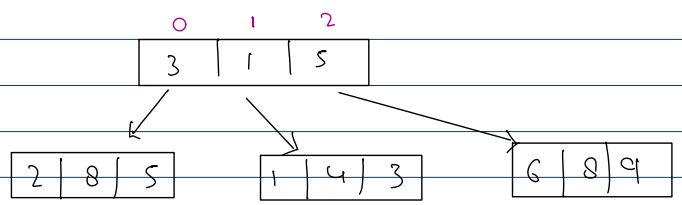
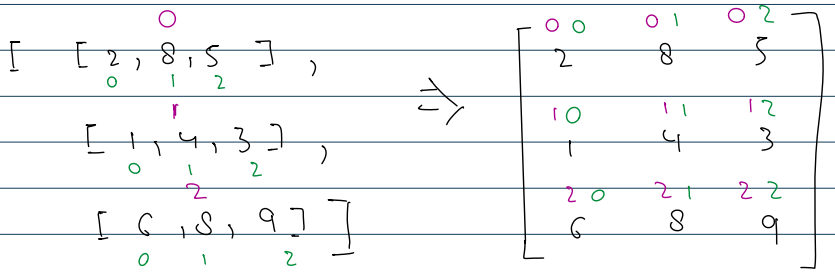
Array

An array is an collection of elements of the same type.

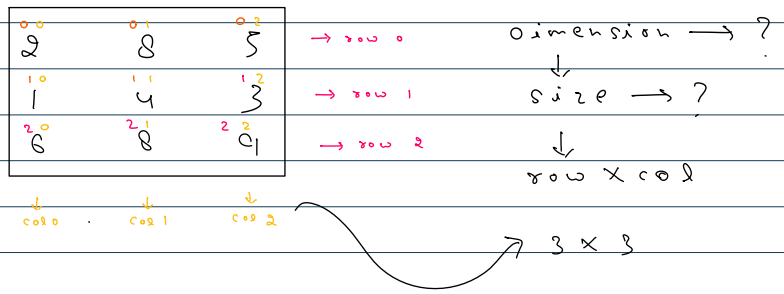
arr ⟶ { 3, 1, 5 } ⟶ 1 D





arr[2] ⇒ [6, 8, 9]

arr[2][0] ⇒ 6



1 D ⟶ int[] arr = new int[5];

2 D ⟶ int[][] arr = new int[3][3];

size

name of array

row

cols

1 D ⟶ int[] arr = { 3, 1, 4 };

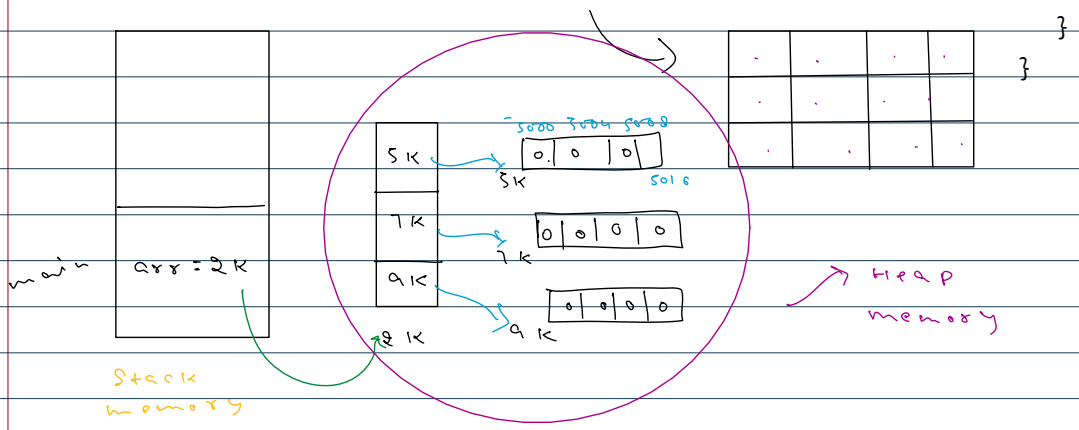
2 D ⟶ int[][] arr = {
{ 1, 2, 3 },
{ 4, 5, 6 },
{ 7, 8, 9 }
}

2D array contiguous memory allocation ?

```
public class Main {
    public static void main(String[] args) {

        int[][] arr = new int[3][4];

        System.out.println(arr);
        System.out.println(arr[0]);

        System.out.println(arr[0][1]);

        int[][] other = arr;
        System.out.println(other);

        int row = arr.length;
        System.out.println(row);
```

```
int col = arr[0].length;
System.out.println(col);
}
}
```

```
int [][] arr = new int [5][4];
```




```
int [][] arr = new int [n][m];
```

n + 1 => Total no. of 1D arrays.

Output of a 2D array

n : 2
m : 3

5	8	9
0	7	3

0 0	0 1	0 2
1 0	1 1	1 2

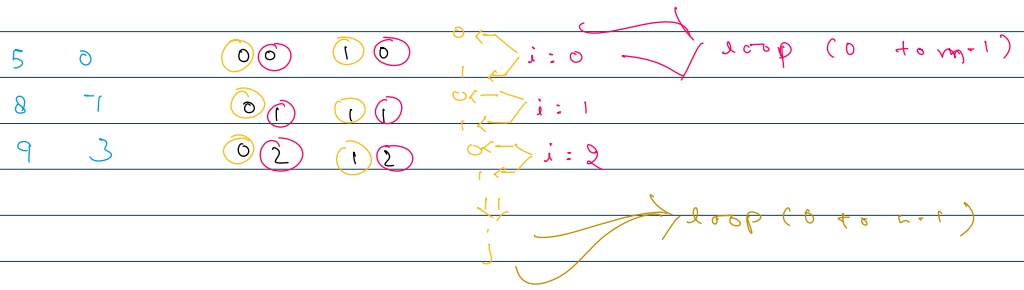
row wise

col wise

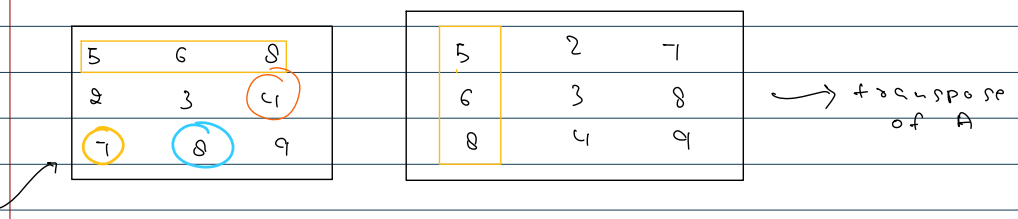
n : 2
m : 3

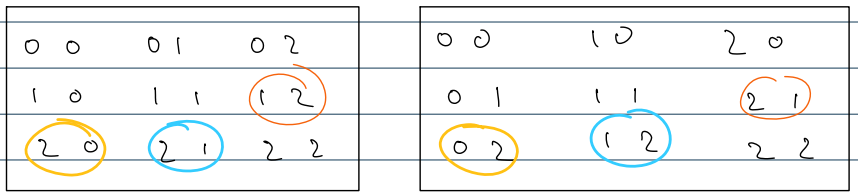
5	8	9
0	7	3

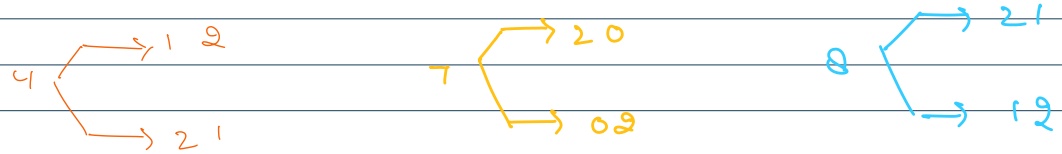
0 0	0 1	0 2
1 0	1 1	1 2



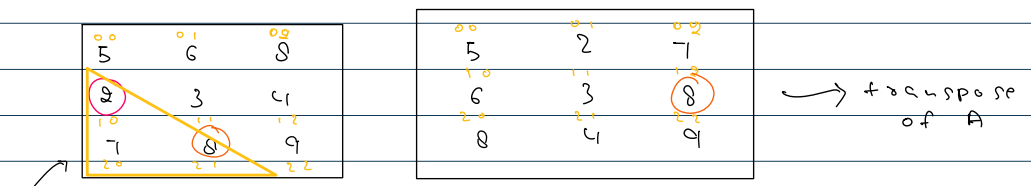
Transpose of a matrix

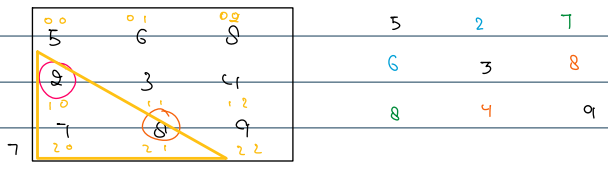






Transpose the matrix in-place





lower triangular matrix → 10 , 20 , 21

→ i > j

```
public class Main {
    public static void main(String[] args) {
```

```
int[][] arr = {
    {5, 8, 9},
    {0, 7, 3},
    {1, 7, 9}
};

int rows = arr.length;
int cols = arr[0].length;

for(int i=0; i<rows; i++){
    for(int j=0; j<cols; j++){
        System.out.print(i+" "+j+ "  ");
    }
    System.out.println();
}

System.out.println("*****************");

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

System.out.println("*****************");

for(int i=0; i<cols; i++){
    for(int j=0; j<rows; j++){
        System.out.print(j+" "+i+ "  ");
    }
    System.out.println();
}

System.out.println("*****************");

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

System.out.println("*****************");

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

```
        }

        for(int i=0; i<rows; i++){
            for(int j=0; j<cols; j++){
                if(i>j)
                swap(arr, i, j);
            }
        }

        System.out.println("*****************");

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


    }

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

}
```

Search in a 2D Matrix II

n → rows
m → cols

Linear Search → O(n*m)

Binary Search on → O(n log m)
every 1D array

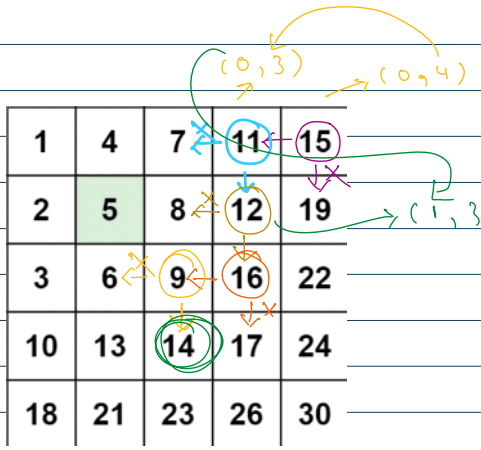
1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

target → 14

```
int row = 0
int col = arr[0].length - 1;

while(_ _ _) {
    if (arr[row][col] == target)
        return true;
    else if (arr[row][col] > target)
        col--;
    else if (arr[row][col] < target)
        row++;
}
```

row ++ ;

}



target = 14

int row = 0

int col = arr[0].length - 1 ;

while ( ) {

if(arr[r][c] == t)

return true ;

else if(arr[r][c] > t)

col - - ;

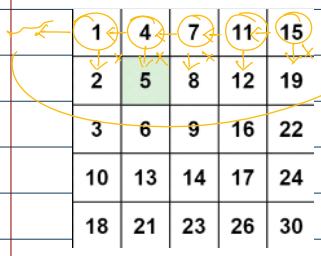
else if (arr[r][c] < t)

row ++ ;

}

target : 25

row == matrix.length



target : 5

col >= 0

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30