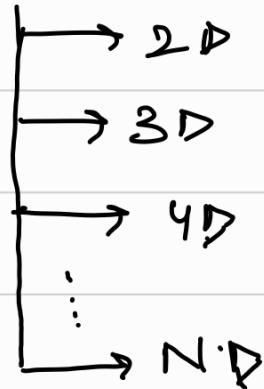


# Arrays

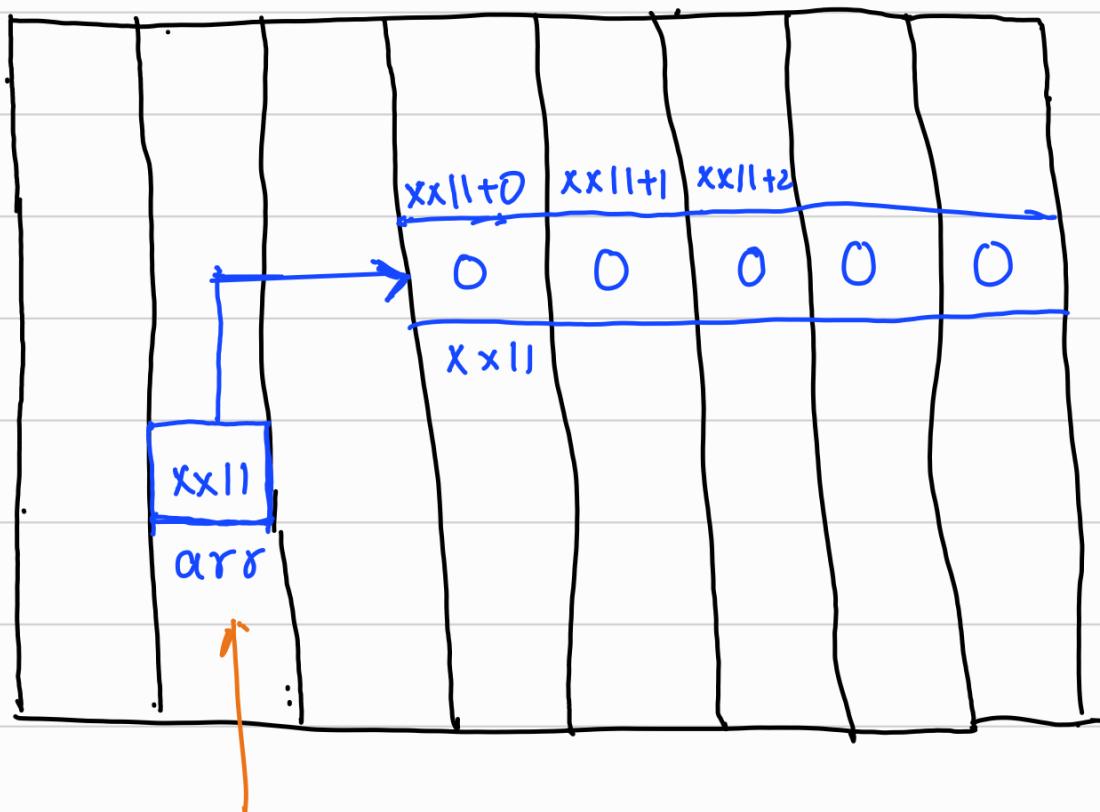
## Arrays

↓  
1D

Multi dimensional.



# Memory



reference in memory → arr

## Syntax → 3 Steps

`int[] intArray;` ---  $O(1)$

datatype Name

ref [ ] no of cells

`intArray = new int[3];` ---  $O(1)$

`intArray[0] = 1;` ---  $O(1)$

`intArray[1] = 2;` ---  $O(1)$

`intArray[2] = 3;` ---  $O(1)$

`System.out[Arrays.toString(intArray)];`

↓

$O(1)$  - String `sArray[] = {"a", "b", "c"};`

`System.out[Arrays.toString(sArray)];`

## Insertion in Array

```
public class IDArray {  
    int arr[] = null;  
  
    constructor  
    public IDArray(int sizeOfArray) {
```

```
arr = new int [size of array];  
for (int i=0 ; i< arr.length ; i++){  
    arr[i] = Integer. Min_value;
```

}

```
public void insert ( int location ,  
                     int value to be  
                     inserted ) {
```

~~try~~

```
if ( arr[location] == Integer. Min_Value )
```

{

```
    arr[location] = value to be  
                    inserted
```

```
SOP ("Success")
```

~~else {~~

~~}~~

```
SOP ("Already occupied")
```

~~catch~~

```
catch (ArrayIndexOutOfBoundsException e) {
```

```
SOP ("Invalid index")
```

}

## Accessing Array Element

### Syntax

`<arrayName>[index]`

Time Complexity  $\rightarrow O(1)$

Space Complexity  $\rightarrow O(1)$

## Array Traversal

Time Complexity  $\rightarrow O(N)$

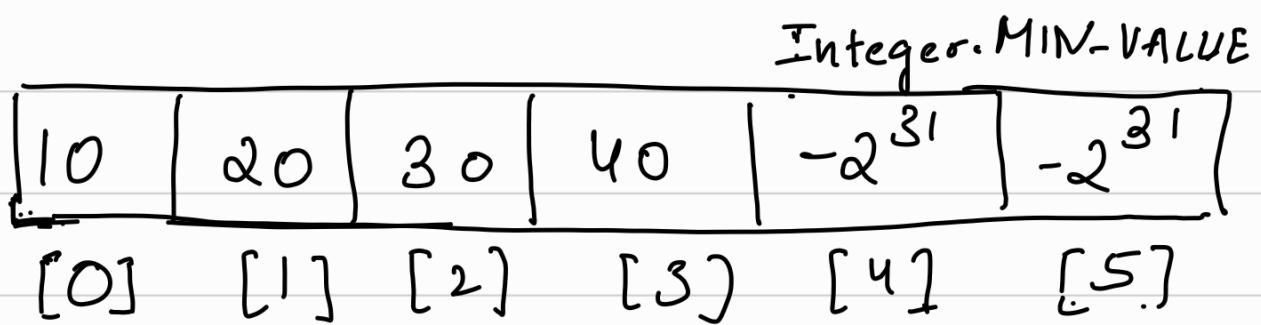
Space Complexity  $\rightarrow O(1)$

## Search for array element

Time Complexity  $\rightarrow O(N)$

Space Complexity  $\rightarrow O(1)$

## Delete Array Element



In java if we delete last 2 elements then also they have some value : Default value = 0 By compiler

Syntax :

`<arrayName>[index] = 2-31`

Time Complexity  $\Rightarrow O(1)$

Space Complexity  $\Rightarrow O(1)$

## 2D Array

Syntax

// Step 1 - Declare

int [] [] int2DArray; --- O(1)

// Step 2 - Instantiate.

int2DArray = new int[2][2]; --- O(1)

// Step 3 - Initialize.

int2DArray [0] [0] = 1

int2DArray [0] [1] = 2

int2DArray [1] [0] = 3

int2DArray [1] [1] = 4

System.out.println(Arrays.deepToString(int2DArray))

String type of array

String s2DArray [][] = { {"a", "b"},  
{ "c", "d"} }

System.out.println(Arrays.deepToString(s2DArray))

Advice => Please do all 3 steps

$\mathcal{L}_m \downarrow$  like it takes  $\mathcal{O}(1)$

Space Complexity  $\rightarrow \mathcal{O}(mn)$

Insertion - 2D Array

Check if each & every cell is occupied or not.

If the cell has value of Integer.MINVALUE or  $2^{-31}$  then we see it as empty & insert a value in it

$\mathcal{O}(1) \rightarrow$  Time complexity

$\mathcal{O}(1) \rightarrow$  Space complexity

Accessing 2D Array

Time complexity  $\rightarrow \mathcal{O}(1)$

Space complexity  $\rightarrow \mathcal{O}(1)$

## Traversal 2D Array.

Time Complexity  $\rightarrow \mathcal{O}(MN)$

Space Complexity  $\rightarrow \mathcal{O}(1)$

## Return 1D Array

```
public class Main {
```

```
    public static int[] gNC() {
```

```
        int[] num = {1, 2, 3, 4, 5};
```

```
        return num;
```

```
}
```

```
psvm {
```

```
    int[] returnArray = gNC();
```

```
    for (int i : returnArray) {
```

```
        System.out.print(i + " ");
```

y

y

## Search in 2D Array

Time complexity  $\rightarrow O(MN)$

Space complexity  $\rightarrow O(1)$

## Delete 2D Array Element

Time complexity  $\rightarrow O(1)$

Space complexity  $\rightarrow O(1)$

