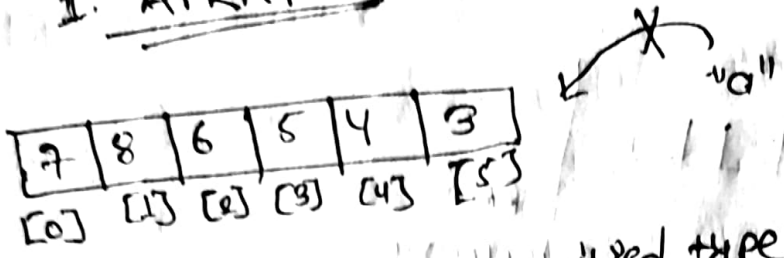
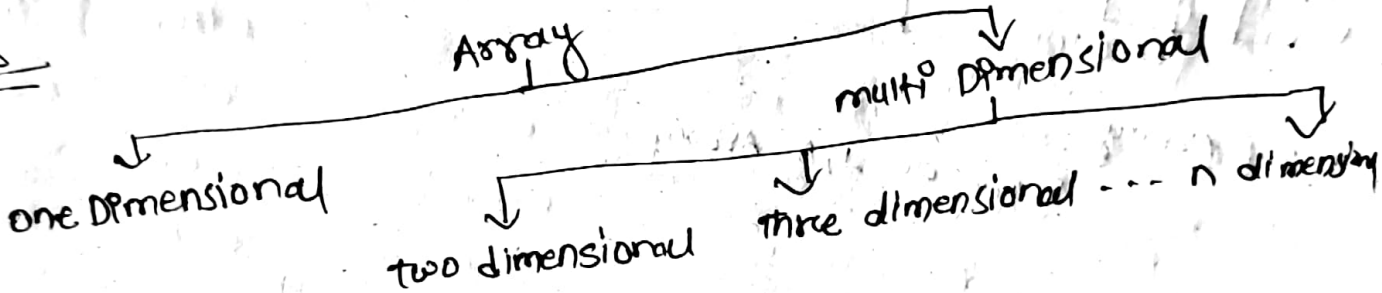
1. ARRAYS



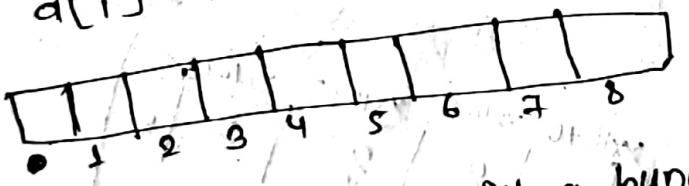
- Array can store data of specified type
- each element has unique index in Array
- elements are stored contiguous in memory
- the size of array is predefined and cannot be modified.

Types



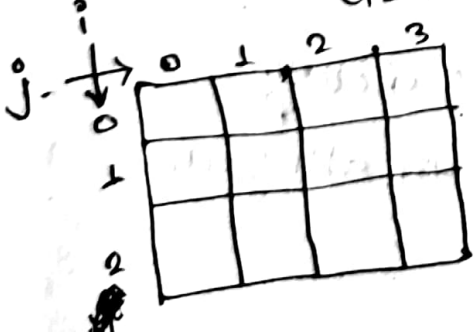
one Dimensional: an array with a bunch of values having been declared with single index.

a[i] → i between 0 and n-1



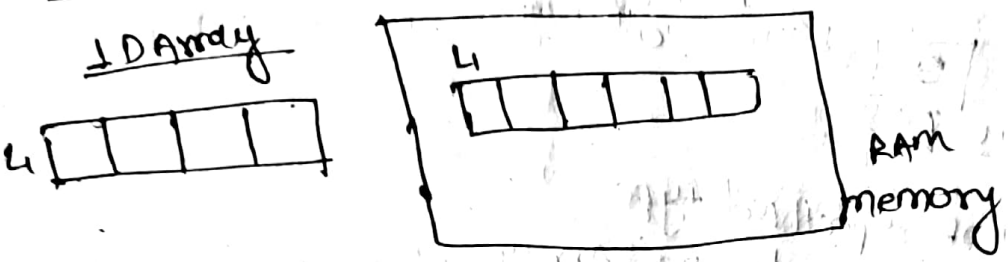
two Dimensional: an array with a bunch of values having been declared with double index.

a[i][j] → i and j between 0 and n

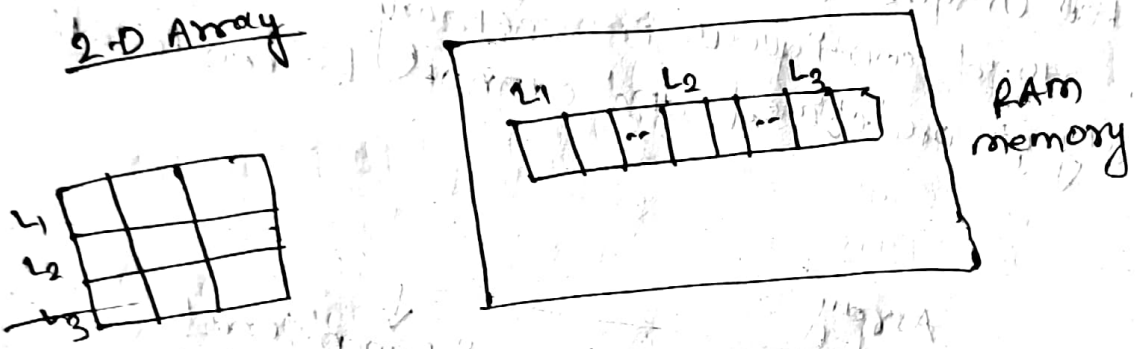


Arrays in memory

1D Array



2D Array

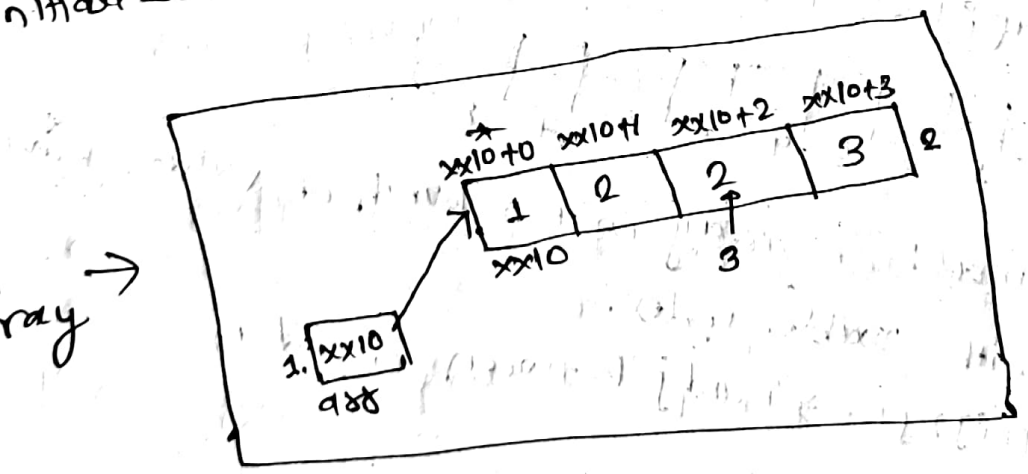


3D Array

same as 1,D Array

Creating an Array

1. Declare - creates a reference to array
2. Instantiation of an array - creates an array
3. initialization - assigns value to cells in array



* arr references to array and store base address of array to access, not to change the base address we start adding from 0 index.

Syntax in Java

1. dataType [] arr
2. arr = new dataType[ ]
3. arr[0] = 2, arr[1] = 3 ...

1-D Array

illustrated in code in pc.

Insertion in Array

code of implementation in pc.

Time complexity = O(1)

Accessing Array Element

O(1) → Time complexity

O(1) → space complexity

Traversal

O(N) → Time complexity

Searching: O(N) → Time complexity

Delete: O(1) → Timeplexity

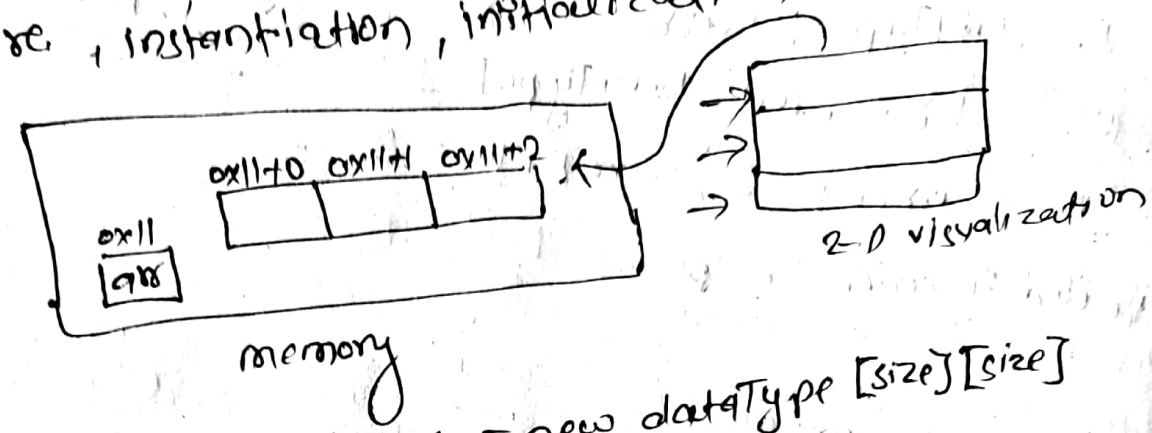
O(1) → Space complexity

Summary

Operation	Time complexity	Space complexity
1. creating an empty array	O(1)	O(n)
2. Inserting a value in array	O(1)	O(1)
3. Traversing a given array	O(N)	O(1)
4. Accessing a given cell	O(1)	O(1)
5. searching a given value	O(N)	
6. Deleting a given value	O(1)	O(1)

Syntax for 2-D array

⇒ Declare, instantiation, initialization



dataType[][] arrayName = new dataType [size][size]

Summary

operation	Time complexity	Space complexity
creating an empty array	O(1)	O(mn)
Inserting a value in array	O(1)	O(1)
Traversing a given array	O(mn)	O(1)
Accessing a given cell	O(1)	O(1)
searching a given value	O(mn)	O(1)
Deleting a given value	O(1)	O(1)

All the above operation is tested and illustrated in pc.

when to use/Avoid Array

Use:
- To store multiple variables of same data type
- Random access

Avoid:
- same data type element
- reverse memory.