Data structures and algorithms

LECTURE

BY

ENGR FATIMA JAFFAR

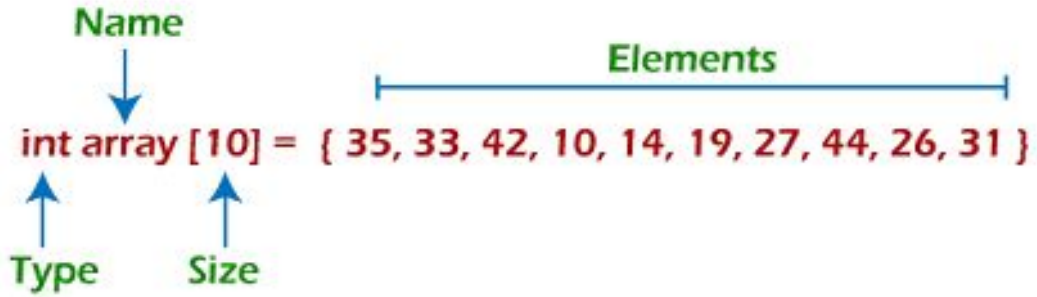
Arrays in DS

- Arrays are defined as the collection of similar types of data items stored at contiguous memory locations.
- It is one of the simplest data structures where each data element can be randomly accessed by using its index number.

Properties of Array

- Each element in an array is of the same data type and carries the same size that is 4 bytes.
- Elements in the array are stored at contiguous memory locations from which the first element is stored at the smallest memory location.
- Elements of the array can be randomly accessed since we can calculate the address of each element of the array with the given base address and the size of the data element.

Representation of array



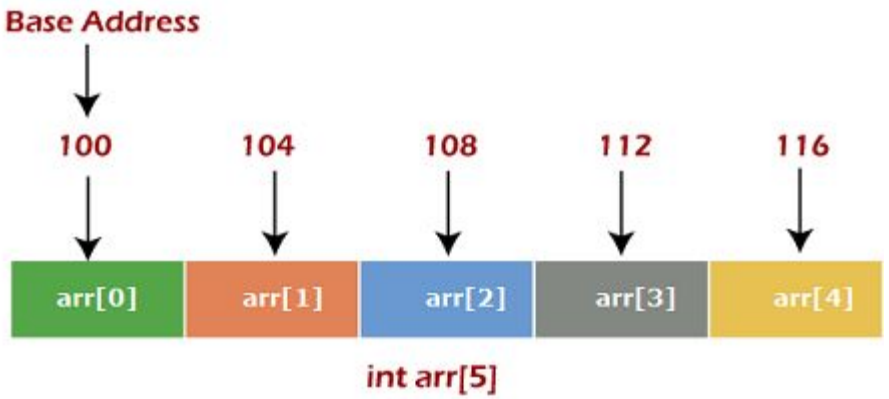
- Index starts with 0.
- The array's length is 10, which means we can store 10 elements.
- Each element in the array can be accessed via its index.

Why are arrays required?

- Sorting and searching a value in an array is easier.
- Arrays are best to process multiple values quickly and easily.
- **Arrays are good for storing multiple values in a single variable**

Memory allocation of array

- data elements of an array are stored at contiguous locations in the main memory.
- The name of the array represents the base address or the address of the first element in the main memory.
- Each element of the array is represented by proper indexing.


Base Address
100
104
108
112
116
arr[0]
arr[1]
arr[2]
arr[3]
arr[4]
int arr[5]

How to access an element in the array

We required the information given below to access any random element from the array

- Base Address of the array.
- Size of an element in bytes.
- Type of indexing, array follows

The formula to calculate the address to access the element.

Byte address of element A[i] = base address + size * (i - first index)

We can understand it with the help of an example -

Suppose an array, A[-10 ..... +2] having Base address (BA) = 999 and size of an element = 2 bytes, find the location of A[-1].

L(A[-1]) = 999 + 2 x [(-1) - (-10)]

= 999 + 18

= 1017

Basic Array Operations

- Traversal - This operation is used to print the elements of the array.
- Insertion - It is used to add an element at a particular index.
- Deletion - It is used to delete an element from a particular index.
- Search - It is used to search an element using the given index or by the value.
- Update - It updates an element at a particular index.

Traversal

This operation is performed to traverse through the array elements. It prints all array elements one after another.

```
#include <iostream>
int main() {
   int Arr[5] = {18, 30, 15, 70, 12};
int i;
   cout<<"Elements of the array are:\n“;
   for(i = 0; i<5; i++) {
      cout<<i<<“ ”<<Arr[i];
Return 0;
   }
}
```

Insertion Operation

This operation is performed to insert one or more elements into the array.

As per the requirements, an element can be added at the beginning, end, or at any index of the array

```java
import java.util.Arrays;
public class Exercise9 {

public static void main(String[] args) {

   int[] my_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

    // Insert an element in 3rd position of the array (index->2, value->5)

   int Index_position = 2;
   int newValue    = 5;

  System.out.println("Original Array : "+Arrays.toString(my_array));

  for(int i=my_array.length-1; i > Index_position; i--){
    my_array[i] = my_array[i-1];
   }
   my_array[Index_position] = newValue;
   System.out.println("New Array: "+Arrays.toString(my_array));
 }
 }
```

Deletion Operation

```
import java.util.Arrays;
public class Main {

    public static void main(String[] args) {
        int[] arr = new int[]{1,2,3,4,5};
        int[] arr_new = new int[arr.length-1];
        int j=3;
        for(int i=0, k=0;i<arr.length;i++){
            if(i!=j){
                arr_new[k]=arr[i];
                k++;
            }
        }
        System.out.println("Before deletion :" + Arrays.toString(arr));
        System.out.println("After deletion :" + Arrays.toString(arr_new));

    }
}
```

Searching

```
import java.util.*;

public class ArrayLinearSearch {
   public static void main(String args[]) {
      int count, num, i;
Scanner in = new Scanner(System.in);
        int arr[10]= {20,56,47,86,45,90,76,45,89,67};
      System.out.println("Enter element to search");
      num = in.nextInt();
      // Compare each element of array with num
      for (i = 0; i < 10 ; i++) {
         if(num == inputArray[i]){
            System.out.println(num+" is present at index "+i);
            break;
         }
      }

      if(i == count)
         System.out.println(num + " not present in input array");
   }
}
```