Arrays

Introduction

Arrays store multiple values of the same datatype in a single variable,making data management easier & cleaner.

Declaring & Initializing an array :

int[] rollNo = new int[5]; → Declaring an array with fixed size.

int[] rollNos = {12, 23, 45, 38, 1}; → Declaring & Initializing an array.

Index :

```
[ 12, 23, 45, 38, 1]
  0   1   2   3   4
```

Array Input :

```
Scanner input = new Scanner(System.in);

int[] arr = new int[4];

for(int i=0; i<arr.length; i++) {

        arr[i] = input.nextInt();

}
```

for - each loop :

Used to iterate through all elements of an array.

```
int[] nums = {10, 20, 30};

for(int n : nums) {

        sout(n);

}
```

toString() :

Converts an object like an array into readable string form.

```
int[] nums = {1, 2, 3};
sout(Arrays.toString(nums));  → [1, 2, 3]
```

Multidimensional Arrays

```
int[][] arr = new int[3][3];
int[][] arr = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};
```

Taking input :

```
Scanner input = new Scanner(System.in);
int[][] arr = new int[3][3];
for(int row=0; row<arr.length; row++) {
    for(int col=0; col<arr[row].length; col++) {
        Arr[row][col] = input.nextInt();
    }
}
```

Printing Output :

```
for(int row=0; row<arr.length; row++) {
        for(int col=0; col<arr[row].length; col++) {
            sout(arr[row][col] + " ");
        }
        sout();
    }
```

Dynamic Array -

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

ArrayList

ArrayList is a dynamic array that can grow or shrink at runtime & allow index-based access.

ArrayList<Integer> list = new Array<>();

1. add() - Adds element to list.

   list.add(5);

2. set() - Replace element at given index.

   list.set(index, value);

3. get() - Fetch element using index.

   sout(list.get(index));

4. remove() - remove element.

list.remove(1); → remove by index

list.remove(Integer.valueOf(5)); → remove by value

5. size() - returns number of elements.

   sout(list.size());

6. contains() - checks if element exists.

   sout(list.contains(25)); → true/false

7. clear() - removes all elements.

   list.clear();

Questions

Easy

1. Largest element in an array.
2. Second largest element in an array.
3. Check if array is sorted (Leetcode - 1752)
4. Remove duplicate from sorted array (Leetcode - 26)
5. Left rotate an array by one place
6. Left rotate an array by D places (Leetcode - 189)
7. Move zeros to end (Leetcode - 283)
8. Linear Search
9. Find Union
10. Find missing number in an array (Leetcode - 268)
11. Maximum Consecutive Ones (Leetcode - 485)
12. Single Number (Leetcode - 136)

Medium

1. Two Sum (Leetcode - 1)
2. Sort Colors : Dutch National Flag (Leetcode - 75)
3. Majority Element (Leetcode - 169)
4. Maximum Subarray Sum : Kadane’s Algorithm (Leetcode - 53)
5. Print subarray with maximum subarray sum (GFG)
6. Best time to buy and sell stocks (Leetcode - 121)
7. Rearrange array elements by sign (Leetcode - 2149)

8. Next Permutation (Leetcode - 31)
9. Container with most water (Leetcode - 11)
10. Longest Consecutive Sequence (Leetcode - 128)
11. Set Matrix Zeroes (Leetcode - 73)
12. Rotate Image (Leetcode - 48)
13. Spiral Matrix (Leetcode - 283)
14. Longest subarray with with given sum K (Leetcode - 560)

Hard

1. Pascal’s Triangle (Leetcode - 118)
2. Majority Element II (Leetcode - 229)
3. 3-Sum (Leetcode - 15)
4. 4-Sum (Leetcode - 18)
5. Largest subarray with 0 sum (GFG)
6. Count number of subarrays with given xor K (InterviewBit)
7. Merge Intervals (Leetcode - 56)
8. Merge Sorted Arrays (Leetcode - 88)
9. Find Repeating and missing numbers (Leetcode - 2965)
10. Count Inversion (Leetcode - 3193)
11. Reverse Pairs (Leetcode - 493)
12. Maximum Product Subarray (Leetcode - 152)