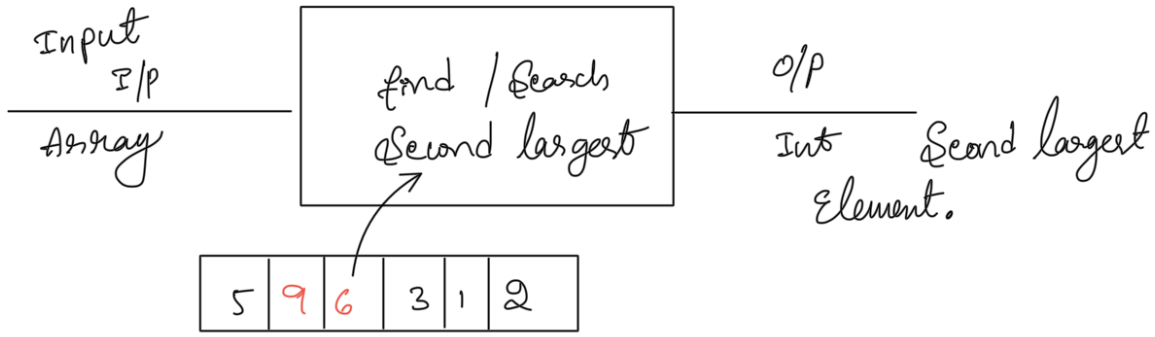
JAVA HANDS ON PRACTICE

Day 24
03/08/25

Second Largest Element In An Array

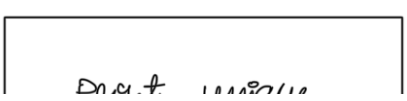


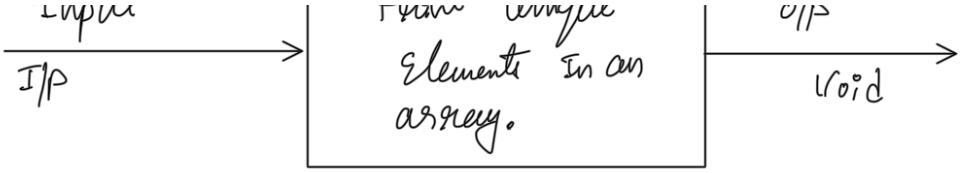
(1) Method 1 : Sort the array and returns Second Largest Element.

(2) Method 2 : Max SecondMax store values acording return SecondMax

```
216
217     public static int findSecoundLarge(int arr[]){
218         if(arr.length<2)return -1;
219         int max = Integer.MIN_VALUE;
220         int smax = Integer.MIN_VALUE;
221         for(int num:arr){
222             if(num>max){
223                 smax=max;
224                 max=num;
225             }
226             else if(num>smax && num<max){
227                 smax=num;
228             }
229         }
230         return smax;
231     }
232
233     public static void invoke_findSecoundLarge() {
234         int[] arr = {10, 5, 8, 20, 20, 15};
235
236         int result = findSecoundLarge(arr);
237         System.out.println("Second Largest Element: " + result);
238     }
239
```

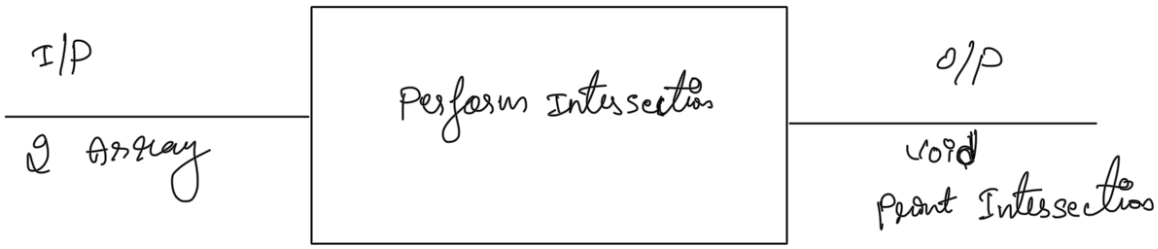
functions to print unique Elements In An Integer Array

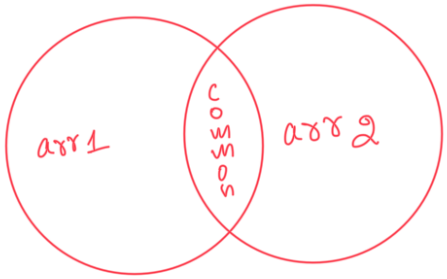





```java
public static void printUnique(int[] arr){
    for(int i = 0;i<arr.length;i++){
        boolean isDuplicate = false ;
        for(int j = 0 ; j < arr.length;j++){
            if(arr[i]==arr[j] && i != j){
                isDuplicate=true;
                break;
            }
        }
    if(!isDuplicate)System.out.print(arr[i]+" ");
    }
}
public static void invoke_printUnique() {
    int[] arr = {4, 5, 3, 4, 2, 5, 7};
    System.out.print(s:"Unique elements: ");
    printUnique(arr);
    System.out.println();
}
Run | Debug | Run main | Debug main
```

Print Common Elements In the Two Array
Intersection





Taking Intersection Helps to find the

```
258    public static void printCommonElements(int[] arr1 , int arr2[]){
259        for(int i = 0; i< arr1.length;i++){
260            boolean isFound = false ;
261            for(int j=0;j<arr2.length;j++){
262                if(arr1[i]==arr2[j]){
263                    isFound=true;
264                    break;
265                }
266            }
267            if(isFound)System.out.println(arr1[i] +" ");
268        }
269
270    }
271
272    public static void invoke_printCommonElements() {
273        int[] arr1 = {1, 2, 3, 4, 5};
274        int[] arr2 = {3, 4, 6, 7, 2};
275
276        System.out.println(x:"Common elements between arr1 and arr2:");
277        printCommonElements(arr1, arr2);
278    }
       Run | Debug | Run main | Debug main
```

Day 24 - Debugging Arrays with Dry Run and Flowcharts

Veeresh K

August 3, 2025

Problem 1: Finding Second Largest Element

Code

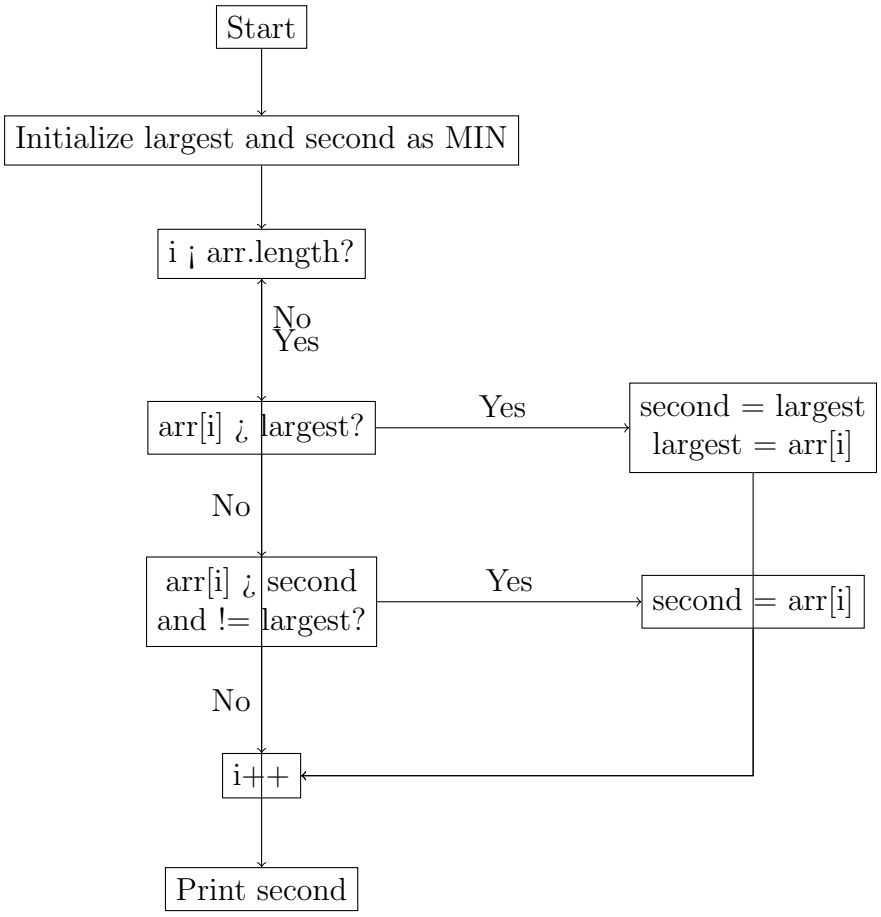
```
int findSecondLargest(int[] arr) {
    int largest = Integer.MIN_VALUE;
    int second = Integer.MIN_VALUE;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] > largest) {
            second = largest;
            largest = arr[i];
        } else if (arr[i] > second && arr[i] != largest) {
            second = arr[i];
        }
    }
    return second;
}
```

Dry Run Example: arr = [5, 3, 9, 1, 6]

- i = 0: arr[i] = 5 ⇒ largest = 5, second = -∞
- i = 1: arr[i] = 3 ⇒ second = 3
- i = 2: arr[i] = 9 ⇒ second = 5, largest = 9
- i = 3: arr[i] = 1 ⇒ (no update)
- i = 4: arr[i] = 6 ⇒ second = 6

Output: 6

Flowchart


Start
Initialize largest and second as MIN
i ¡ arr.length?
No
Yes
arr[i] ¿ largest?
Yes
second = largest
largest = arr[i]
No
arr[i] ¿ second
and != largest?
Yes
second = arr[i]
No
i++
Print second

Problem 2: Unique Elements in Array

Code

```
void printUnique(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        boolean isUnique = true;
        for (int j = 0; j < arr.length; j++) {
            if (arr[i] == arr[j] && i != j) {
                isUnique = false;
                break;
            }
        }
        if (isUnique) System.out.println(arr[i]);
    }
}
```

Dry Run: arr = [3, 4, 5, 3, 6]

- i = 0: arr[0] = 3 ⇒ Found match at j = 3 ⇒ $Notuniquei = 1 : arr[1] = 4$⇒ Unique
- i = 2: arr[2] = 5 ⇒ Unique
- i = 3: arr[3] = 3 ⇒ Match with i=0 ⇒ $Notuniquei = 4 : arr[4] = 6$⇒ Unique

Output: 4 5 6

Problem 3: Print Common Elements of Two Arrays

Code

```
void printCommon(int[] a, int[] b) {
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < b.length; j++) {
            if (a[i] == b[j]) {
                System.out.println(a[i]);
                break;
            }
        }
    }
}
```

Dry Run: a = [1, 2, 3], b = [3, 4, 1]

- a[0]=1 ⇒ Match with b[2] ⇒ $Print1a[1] = 2$⇒ No match
- a[2]=3 ⇒ Match with b[0] ⇒ $Print3$ **Output:** 1 3