***Second Largest Element in Array***

Given an array of integers, our task is to write a program that efficiently finds the second largest element present in the array.

**Example:**

**Input:** arr[] = {12, 35, 1, 10, 34, 1}

**Output:** The second largest element is 34.

**Explanation:** The largest element of the

array is 35 and the second

largest element is 34

**Input:** arr[] = {10, 5, 10}

**Output:** The second largest element is 5.

**Explanation:** The largest element of

the array is 10 and the second

largest element is 5

**Input:** arr[] = {10, 10, 10}

**Output:** The second largest does not exist.

**Explanation:** Largest element of the array

is 10 there is no second largest element

**Efficient Solution:**

**Approach:** Find the second largest element in a single traversal.   
Below is the complete algorithm for doing this:

1) Initialize the first as 0(i.e, index of arr[0] element

2) Start traversing the array from array[1],

a) If the current element in array say arr[i] is greater

than first. Then update first and second as,

second = first

first = arr[i]

b) If the current element is in between first and second,

then update second to store the value of current variable as

second = arr[i]

3) Return the value stored in second.

**Implementation:**

C++Java

// JAVA Code for Find Second largest

// element in an array

class GFG {

/\* Function to print the second largest

elements \*/

public static void print2largest(int arr[],

int arr\_size)

{

int i, first, second;

/\* There should be atleast two elements \*/

if (arr\_size < 2) {

System.out.print(" Invalid Input ");

return;

}

first = second = Integer.MIN\_VALUE;

for (i = 0; i < arr\_size; i++) {

/\* If current element is greater than

first then update both first and second \*/

if (arr[i] > first) {

second = first;

first = arr[i];

}

/\* If arr[i] is in between first and

second then update second \*/

else if (arr[i] > second && arr[i] != first)

second = arr[i];

}

if (second == Integer.MIN\_VALUE)

System.out.print("There is no second largest"

+ " element\n");

else

System.out.print("The second largest element"

+ " is " + second);

}

/\* Driver program to test above function \*/

public static void main(String[] args)

{

int arr[] = { 12, 35, 1, 10, 34, 1 };

int n = arr.length;

print2largest(arr, n);

}

}

**Output**

The second largest element is 34

**Complexity Analysis:**

* **Time Complexity: O(n).**  
  Only one traversal of the array is needed.
* **Auxiliary space:** **O(1).**  
  As no extra space is required.